electronics

Special Issue Reprint

Feature Papers in Artificial Intelligence

Edited by
Xin Geng, George A. Papakostas, Fernando De la Prieta Pintado
and Valentina Emilia Balas

mdpi.com/journal/electronics

MDPI

Feature Papers in Artificial Intelligence

Feature Papers in Artificial Intelligence

Guest Editors

Xin Geng
George A. Papakostas
Fernando De la Prieta Pintado
Valentina Emilia Balas

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Xin Geng
School of Computer Science and Engineering
Southeast University
Nanjing
China

George A. Papakostas
Department of Informatics
Democritus University of Thrace
Kavala
Greece

Fernando De la Prieta Pintado
BISITE Research Group
University of Salamanca
Salamanca
Spain

Valentina Emilia Balas
Department of Automatics and Applied Software
Aurel Vlaicu University of Arad
Arad
Romania

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Electronics* (ISSN 2079-9292), freely accessible at: https://www.mdpi.com/journal/electronics/special_issues/R0PEBM750D.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-8789-7 (Hbk)
ISBN 978-3-7258-8790-3 (PDF)
https://doi.org/10.3390/books978-3-7258-8790-3

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

Fernando De la Prieta Pintado, George A. Papakostas, Valentina E. Balas and Xin Geng
Feature Papers in Artificial Intelligence
Reprinted from: *Electronics* **2026**, *15*, 2841, https://doi.org/10.3390/electronics15132841 . . . . . **1**

J. Ernesto Solanes and Aitana Francés-Falip
Geometry-Aware Conformal Calibration of Entropic Soft-Min Operators for Machine Learning and Reinforcement Learning
Reprinted from: *Electronics* **2026**, *15*, 1704, https://doi.org/10.3390/electronics15081704 . . . . . **5**

Jiwoo Lee and Eun-Kyu Lee
Information-Preserving Spiking for Accurate Time-Series Forecasting in Spiking Neural Networks
Reprinted from: *Electronics* **2026**, *15*, 1597, https://doi.org/10.3390/electronics15081597 . . . . . **39**

Alejandro J. González-Santana, Giovanny A. Cuervo-Londoño and Javier Sánchez
Ensemble Graph Neural Networks for Probabilistic Sea Surface Temperature Forecasting via Input Perturbations
Reprinted from: *Electronics* **2026**, *15*, 1583, https://doi.org/10.3390/electronics15081583 . . . . . **57**

Andrea Caridi, Marco Giovannini and Lorenzo Ricciardi Celsi
AI-Assisted Value Investing: A Human-in-the-Loop Framework for Prompt-Guided Financial Analysis and Decision Support
Reprinted from: *Electronics* **2026**, *15*, 1155, https://doi.org/10.3390/electronics15061155 . . . . . **86**

Fatima Saidu and Julie Wall
Retrieval-Augmented Large Language Model for Clinical Decision Support with a Medical Knowledge Graph
Reprinted from: *Electronics* **2026**, *15*, 555, https://doi.org/10.3390/electronics15030555 . . . . . **111**

Hisatoshi Naganawa, Enna Hirata and Akira Yamada
Implementing a Knowledge Management System with GraphRAG: A Physical Internet Example
Reprinted from: *Electronics* **2025**, *14*, 4948, https://doi.org/10.3390/electronics14244948 . . . . . **137**

Giovanny A. Cuervo-Londoño, José G. Reyes, Ángel Rodríguez-Santana and Javier Sánchez
Voronoi-Induced Artifacts from Grid-to-Mesh Coupling and Bathymetry-Aware Meshes in Graph Neural Networks for Sea Surface Temperature Forecasting
Reprinted from: *Electronics* **2025**, *14*, 4841, https://doi.org/10.3390/electronics14244841 . . . . . **156**

Chien-Yu Lu, Hong-Yi Hsu, Bo-Siang Chen, Wei-Lun Huang and Wei-Sho Ho
Development and Validation of an Explainable Hybrid Deep Learning Model for Multiple-Fault Diagnosis in Intelligent Automotive Electronic Systems
Reprinted from: *Electronics* **2025**, *14*, 4488, https://doi.org/10.3390/electronics14224488 . . . . . **183**

Manal Jammal, Javier Parra Domínguez, Laura Grande-Pérez and Fernando De la Prieta Pintado
Unsupervised Learning of Energy States in Automated Storage Systems with Self-Organizing Maps
Reprinted from: *Electronics* **2025**, *14*, 4365, https://doi.org/10.3390/electronics14224365 . . . . . **201**

Simone Volturno, Andrea Di Martino and Michela Longo
Machine Learning-Based State-of-Charge Prediction for Electric Bus Fleet: A Critical Analysis
Reprinted from: *Electronics* **2025**, *14*, 4147, https://doi.org/10.3390/electronics14214147 . . . . . **217**

Sua Kang, Chaelim Lee, Subin Jung and Minsu Lee
Fic2Bot: A Scalable Framework for Persona-Driven Chatbot Generation from Fiction
Reprinted from: *Electronics* **2025**, *14*, 3859, https://doi.org/10.3390/electronics14193859 . . . . . **234**

Jisoo Baik, Youngbeom Jung and Kyujoong Lee
Frequency-Aware Residual Networks with Masked Dual-Path Convolution for Image Classification
Reprinted from: *Electronics* **2025**, *14*, 3690, https://doi.org/10.3390/electronics14183690 . . . . . **259**

Yu Du, Jiaju Shi, Xun Dou and Yu He
A Short-Term User-Side Load Forecasting Method Based on the MCPO-VMD-FDFE Decomposition-Enhanced Framework
Reprinted from: *Electronics* **2025**, *14*, 3611, https://doi.org/10.3390/electronics14183611 . . . . . **270**

Esteban García-Cuesta, Daniel Manrique and Radu Constantin Ionescu
Interpretable Deep Prototype-Based Neural Networks: Can a 1 Look like a 0?
Reprinted from: *Electronics* **2025**, *14*, 3584, https://doi.org/10.3390/electronics14183584 . . . . . **293**

Javier Sevilla-Salcedo, José Carlos Castillo Montoya, Álvaro Castro-González and Miguel A. Salichs
From Games to Understanding: Semantrix as a Testbed for Advancing Semantics in Human–Computer Interaction with Transformers
Reprinted from: *Electronics* **2025**, *14*, 3480, https://doi.org/10.3390/electronics14173480 . . . . . **309**

Yue Zuo, Ning Ning, Ke Cao, Rui Zhang, Cheng Fu, Shengxin Wang, et al.
High-Density Neuromorphic Inference Platform (HDNIP) with 10 Million Neurons
Reprinted from: *Electronics* **2025**, *14*, 3412, https://doi.org/10.3390/electronics14173412 . . . . . **340**

Constantin-Bogdan Popescu, Laura Florea and Corneliu Florea
Mitigating Context Bias in Vision–Language Models via Multimodal Emotion Recognition
Reprinted from: *Electronics* **2025**, *14*, 3311, https://doi.org/10.3390/electronics14163311 . . . . . **362**

Abdulrahman M. Abdulghani, Azizol Abdullah, Amir Rizaan Rahiman, Nor Asilah Wati Abdul Hamid and Bilal Omar Akram
Network-Aware Gaussian Mixture Models for Multi-Objective SD-WAN Controller Placement
Reprinted from: *Electronics* **2025**, *14*, 3044, https://doi.org/10.3390/electronics14153044 . . . . . **389**

SeongYeon Oh, Moonryul Jung and Taehoon Kim
EnvMat: A Network for Simultaneous Generation of PBR Maps and Environment Maps from a Single Image
Reprinted from: *Electronics* **2025**, *14*, 2554, https://doi.org/10.3390/electronics14132554 . . . . . **422**

Kyriakos D. Apostolidis and George A. Papakostas
Delving into YOLO Object Detection Models: Insights into Adversarial Robustness
Reprinted from: *Electronics* **2025**, *14*, 1624, https://doi.org/10.3390/electronics14081624 . . . . . **457**

Leonidas Theodorakopoulos, Alexandra Theodoropoulou and Constantinos Halkiopoulos
Cognitive Bias Mitigation in Executive Decision-Making: A Data-Driven Approach Integrating Big Data Analytics, AI, and Explainable Systems
Reprinted from: *Electronics* **2025**, *14*, 3930, https://doi.org/10.3390/electronics14193930 . . . . . **492**

Mahmoud Z. A. Wahba, Sara Baldoni and Federica Battisti
Learning-Based Viewport Prediction for 360-Degree Videos: A Review
Reprinted from: *Electronics* **2025**, *14*, 3743, https://doi.org/10.3390/electronics14183743 . . . . . **544**

electronics

MDPI

Editorial

Feature Papers in Artificial Intelligence

Fernando De la Prieta Pintado [1,*], George A. Papakostas [2], Valentina E. Balas [3] and Xin Geng [4]

[1] BISITE Research Group, Department of Computer Science, University of Salamanca, Edificio Multiusos I+D+I, C/Espejos s/n, 37008 Salamanca, Spain
[2] MLV Research Group, Department of Computer Science, Democritus University of Thrace, Agios Loukas, 65404 Kavala, Greece; gpapak@cs.duth.gr
[3] Department of Automation and Applied Informatics, Aurel Vlaicu University of Arad, 310130 Arad, Romania; valentina.balas@uav.ro
[4] School of Computer Science and Engineering, Southeast University, Nanjing 211189, China; xgeng@seu.edu.cn
* Correspondence: fer@usal.es; Tel.: +34-677-522-678

1. Introduction

Artificial intelligence has revolutionised numerous aspects of our society. Foundation models and embodied intelligence are expanding the frontiers of AI research, and the convergence of AI with fundamental sciences has accelerated scientific discovery across disciplines. This Feature Paper Special Issue was put together to showcase research that advances both the theoretical understanding and the practical application of AI technologies.

The call welcomed contributions across the full spectrum of AI research, from novel methodologies and theoretical frameworks to transformative applications. The scope also covered research on AI system development challenges—scaling approaches, low-cost deployment, system reliability, trustworthiness, and human–AI interaction paradigms—as well as submissions that bridge AI with other scientific fields, such as drug discovery and materials science. After the standard peer-review process, twenty-two papers were accepted and published in the Special Issue [Contributions 1–22] by groups based in Europe, Asia, and the Americas. The collection includes original research articles, two reviews, and several application papers.

Rather than forcing a single narrative, we briefly summarise the contributions below in the order of the main themes the issue ended up covering.

2. The Present Issue

Several papers deal with computer vision and multimodal perception. Apostolidis and Papakostas [Contribution 1] revisit the YOLO family of object detectors and study how their accuracy degrades under adversarial attacks and image corruption, a question that matters now that YOLO variants are routinely deployed in safety-critical settings. Baik et al. [Contribution 2] propose a residual network with masked dual-path convolution that explicitly separates high- and low-frequency information for image classification. Oh, Jung and Kim [Contribution 3] tackle the joint generation of physically based rendering and environment maps using a diffusion-based architecture (EnvMat), with immersive 3D content as the main target. Popescu, Florea and Florea [Contribution 4] focus on emotion recognition with vision–language models and show, using carefully constructed synthetic datasets, that VLMs are heavily influenced by the background of a scene; they then propose a multimodal method to mitigate this bias. The issue also includes a review by Wahba, Baldoni and Battisti [Contribution 5] on learning-based viewport prediction for 360-degree video, which points at the gaps that the next generation of systems will need to close.

Language models and human-centred AI are the focus of another set of contributions. Sevilla-Salcedo et al. [Contribution 6] use a semantic word-guessing platform (Semantrix) to compare classical and Transformer-based embeddings under realistic user interaction; the 2×2 factorial design with real participants is worth noting, since it goes beyond the usual benchmark comparison. Kang et al. [Contribution 7] present Fic2Bot, a pipeline that uses LLMs and retrieval-augmented generation to derive persona-driven chatbots from works of fiction. Naganawa, Hirata and Yamada [Contribution 8] apply GraphRAG to a knowledge management problem in the physical internet, illustrating where graph-based retrieval pays off compared to plain RAG. Saidu and Wall [Contribution 9] couple a medical knowledge graph with an LLM for clinical decision support, paying particular attention to the traceability of the answers—a non-negotiable property in this domain. Caridi, Giovannini and Ricciardi Celsi [Contribution 10] explore the use of LLMs in value investing through a human-in-the-loop framework, and the paper is also a sober reading on the model risks involved.

Three contributions look at trustworthy and interpretable AI. García-Cuesta, Manrique and Ionescu [Contribution 11] take a critical view of prototype-based neural networks and show that the prototypes, normally presented as inherently interpretable, can be manipulated without losing the appearance of explainability. Lu et al. [Contribution 12] develop an explainable hybrid deep learning model for fault diagnosis in automotive electronic systems, where explanation is not a nice-to-have but a regulatory requirement. Theodorakopoulos, Theodoropoulou and Halkiopoulos [Contribution 13] step back and look at cognitive bias mitigation in executive decision-making, discussing both what XAI can do and what organisational barriers still get in the way.

Two papers come from the brain-inspired side of the field. Zuo et al. [Contribution 14] describe HDNIP, a high-density neuromorphic inference platform that combines a density-first architecture with a ReRAM-based near-memory strategy and reaches a neuron density two orders of magnitude above prior art. Lee and Lee [Contribution 15] tackle an often-overlooked problem of spiking neural networks in time-series tasks—the information loss introduced by spike encoding—and propose an information-preserving scheme.

The largest single group of papers applies AI to forecasting, energy, industrial systems, and networks. Du et al. [Contribution 16] design a short-term user-side load forecasting method that combines variational mode decomposition with frequency-domain features and multi-scale convolution. Volturno, Di Martino and Longo [Contribution 17] offer a critical comparison of machine learning models for state-of-charge prediction in electric bus fleets, paying attention to the data limitations of real deployments. Jammal et al. [Contribution 18] use Self-Organising Maps to learn energy states of automated storage and retrieval systems in an unsupervised manner. Sea surface temperature forecasting is covered in two complementary papers: Cuervo-Londoño et al. [Contribution 19] identify Voronoi-induced artefacts when graph neural networks are coupled to physical grids and propose bathymetry-aware meshes, while González-Santana, Cuervo-Londoño and Sánchez [Contribution 20] move to probabilistic forecasting using ensemble GNNs and structured input perturbations. The group closes with Abdulghani et al. [Contribution 21], who propose a network-aware Gaussian Mixture Model for controller placement in SD-WAN deployments.

Closer to the theoretical side, Solanes and Francés-Falip [Contribution 22] propose a geometry-aware conformal calibration of entropic soft-min operators with distribution-free guarantees, with applications in both machine learning and reinforcement learning. It is the kind of foundational work that will quietly benefit a number of applied papers downstream.

https://doi.org/10.3390/electronics15132841

3. Conclusions

The papers collected here cover a wide stretch of AI research, from neuromorphic chips to the social consequences of biased decision support, and from low-level signal processing to language-model-driven reasoning. What we noticed across most of them, while reading them in preparation for this editorial, is a clear preoccupation with the conditions under which AI systems actually work in practice—data, robustness, interpretability, and deployment constraints—rather than with benchmark scores alone. We think this is a healthy direction. Our hope is that the Special Issue will be a useful starting point for readers entering any of these topics, and that some of the open questions raised by the authors will end up shaping future work.

Author Contributions: F.D.l.P.P., G.A.P., V.E.B. and X.G. worked together throughout the editorial process of the Special Issue "Feature Papers in Artificial Intelligence" in the journal *Electronics*. F.D.l.P.P. drafted this editorial; G.A.P., V.E.B. and X.G. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This editorial received no specific external funding.

Acknowledgments: We thank all the authors who contributed to this Special Issue, as well as the reviewers whose careful work made the published papers stronger. We are also grateful to the editorial board of *Electronics* for the invitation to guest-edit this Special Issue, and to the editorial office for their support throughout the process.

Conflicts of Interest: The authors declare no conflicts of interest.

List of Contributions:

1. Apostolidis, K.D.; Papakostas, G.A. Delving into YOLO Object Detection Models: Insights into Adversarial Robustness. *Electronics* **2025**, *14*, 1624. https://doi.org/10.3390/electronics14081624.
2. Baik, J.; Jung, Y.; Lee, K. Frequency-Aware Residual Networks with Masked Dual-Path Convolution for Image Classification. *Electronics* **2025**, *14*, 3690. https://doi.org/10.3390/electronics14183690.
3. Oh, S.; Jung, M.; Kim, T. EnvMat: A Network for Simultaneous Generation of PBR Maps and Environment Maps from a Single Image. *Electronics* **2025**, *14*, 2554. https://doi.org/10.3390/electronics14132554.
4. Popescu, C.-B.; Florea, L.; Florea, C. Mitigating Context Bias in Vision–Language Models via Multimodal Emotion Recognition. *Electronics* **2025**, *14*, 3311. https://doi.org/10.3390/electronics14163311.
5. Wahba, M.Z.A.; Baldoni, S.; Battisti, F. Learning-Based Viewport Prediction for 360-Degree Videos: A Review. *Electronics* **2025**, *14*, 3743. https://doi.org/10.3390/electronics14183743.
6. Sevilla-Salcedo, J.; Castillo Montoya, J.C.; Castro-González, Á.; Salichs, M.A. From Games to Understanding: Semantrix as a Testbed for Advancing Semantics in Human–Computer Interaction with Transformers. *Electronics* **2025**, *14*, 3480. https://doi.org/10.3390/electronics14173480.
7. Kang, S.; Lee, C.; Jung, S.; Lee, M. Fic2Bot: A Scalable Framework for Persona-Driven Chatbot Generation from Fiction. *Electronics* **2025**, *14*, 3859. https://doi.org/10.3390/electronics14193859.
8. Naganawa, H.; Hirata, E.; Yamada, A. Implementing a Knowledge Management System with GraphRAG: A Physical Internet Example. *Electronics* **2025**, *14*, 4948. https://doi.org/10.3390/electronics14244948.
9. Saidu, F.; Wall, J. Retrieval-Augmented Large Language Model for Clinical Decision Support with a Medical Knowledge Graph. *Electronics* **2026**, *15*, 555. https://doi.org/10.3390/electronics15030555.
10. Caridi, A.; Giovannini, M.; Ricciardi Celsi, L. AI-Assisted Value Investing: A Human-in-the-Loop Framework for Prompt-Guided Financial Analysis and Decision Support. *Electronics* **2026**, *15*, 1155. https://doi.org/10.3390/electronics15061155.

11. García-Cuesta, E.; Manrique, D.; Ionescu, R.C. Interpretable Deep Prototype-Based Neural Networks: Can a 1 Look like a 0? *Electronics* **2025**, *14*, 3584. https://doi.org/10.3390/electronics14183584.
12. Lu, C.-Y.; Hsu, H.-Y.; Chen, B.-S.; Huang, W.-L.; Ho, W.-S. Development and Validation of an Explainable Hybrid Deep Learning Model for Multiple-Fault Diagnosis in Intelligent Automotive Electronic Systems. *Electronics* **2025**, *14*, 4488. https://doi.org/10.3390/electronics14224488.
13. Theodorakopoulos, L.; Theodoropoulou, A.; Halkiopoulos, C. Cognitive Bias Mitigation in Executive Decision-Making: A Data-Driven Approach Integrating Big Data Analytics, AI, and Explainable Systems. *Electronics* **2025**, *14*, 3930. https://doi.org/10.3390/electronics14193930.
14. Zuo, Y.; Ning, N.; Cao, K.; Zhang, R.; Fu, C.; Wang, S.; Meng, L.; Ma, R.; Qiao, G.; Liu, Y.; et al. High-Density Neuromorphic Inference Platform (HDNIP) with 10 Million Neurons. *Electronics* **2025**, *14*, 3412. https://doi.org/10.3390/electronics14173412.
15. Lee, J.; Lee, E.-K. Information-Preserving Spiking for Accurate Time-Series Forecasting in Spiking Neural Networks. *Electronics* **2026**, *15*, 1597. https://doi.org/10.3390/electronics15081597.
16. Du, Y.; Shi, J.; Dou, X.; He, Y. A Short-Term User-Side Load Forecasting Method Based on the MCPO-VMD-FDFE Decomposition-Enhanced Framework. *Electronics* **2025**, *14*, 3611. https://doi.org/10.3390/electronics14183611.
17. Volturno, S.; Di Martino, A.; Longo, M. Machine Learning-Based State-of-Charge Prediction for Electric Bus Fleet: A Critical Analysis. *Electronics* **2025**, *14*, 4147. https://doi.org/10.3390/electronics14214147.
18. Jammal, M.; Parra Domínguez, J.; Grande-Pérez, L.; De la Prieta Pintado, F. Unsupervised Learning of Energy States in Automated Storage Systems with Self-Organizing Maps. *Electronics* **2025**, *14*, 4365. https://doi.org/10.3390/electronics14224365.
19. Cuervo-Londoño, G.A.; Reyes, J.G.; Rodríguez-Santana, Á.; Sánchez, J. Voronoi-Induced Artifacts from Grid-to-Mesh Coupling and Bathymetry-Aware Meshes in Graph Neural Networks for Sea Surface Temperature Forecasting. *Electronics* **2025**, *14*, 4841. https://doi.org/10.3390/electronics14244841.
20. González-Santana, A.J.; Cuervo-Londoño, G.A.; Sánchez, J. Ensemble Graph Neural Networks for Probabilistic Sea Surface Temperature Forecasting via Input Perturbations. *Electronics* **2026**, *15*, 1583. https://doi.org/10.3390/electronics15081583.
21. Abdulghani, A.M.; Abdullah, A.; Rahiman, A.R.; Abdul Hamid, N.A.W.; Akram, B.O. Network-Aware Gaussian Mixture Models for Multi-Objective SD-WAN Controller Placement. *Electronics* **2025**, *14*, 3044. https://doi.org/10.3390/electronics14153044.
22. Solanes, J.E.; Francés-Falip, A. Geometry-Aware Conformal Calibration of Entropic Soft-Min Operators for Machine Learning and Reinforcement Learning. *Electronics* **2026**, *15*, 1704. https://doi.org/10.3390/electronics15081704.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Geometry-Aware Conformal Calibration of Entropic Soft-Min Operators for Machine Learning and Reinforcement Learning

J. Ernesto Solanes [1,*] and Aitana Francés-Falip [2]

[1] Instituto de Diseño y Fabricación, Universitat Politècnica de València, 46022 València, Spain
[2] Digital Integrated Technologies and E-Health, Technological Institute for Children's Products and Leisure (AIJU), 03440 Alicante, Spain
* Correspondence: esolanes@idf.upv.es

Abstract

Entropic soft-min operators are widely used to obtain smooth approximations of minimum and argmin mechanisms in optimization, machine learning, and reinforcement learning. The quality of this approximation is controlled by an inverse temperature parameter that governs the trade-off between smoothness and fidelity, yet its selection is usually based on global heuristics or worst-case bounds that do not account for the geometry of the candidate cost vector. This study investigates the calibration of the inverse temperature parameter from a geometry-aware perspective, with explicit guarantees on the approximation error between the entropic soft-min and the exact minimum value. After establishing the structural properties of the relaxation error, including monotonicity with respect to the inverse temperature and its dependence on the geometry of the near-optimal set, we introduce a conformal calibration rule that selects the smallest inverse temperature, ensuring that a prescribed upper quantile of the approximation error remains below a target tolerance with distribution-free finite-sample validity. The resulting selector adapts to the geometry distribution represented in the calibration population and provides a principled alternative to mean-based and worst-case tuning rules. Numerical experiments, including geometry-controlled benchmarks and a contextual bandit setting illustrating the impact of geometry-aware calibration on decision-making under estimated action values, show that the proposed method accurately tracks oracle calibration temperatures, preserves the desired operator-level coverage, and makes explicit how geometric heterogeneity governs the effective sharpness required by the soft-min approximation. Additional shifted evaluations illustrate the role of exchangeability in the validity guarantee and the consequences of transferring temperatures across populations with different near-optimal geometries.

Keywords: entropic soft-min relaxation; conformal calibration; distribution-free guarantees; parameter selection in optimization; log-sum-exp approximation

1. Introduction

Smooth approximations of non-smooth operators play a central role in contemporary optimization, machine learning, and control. Many architectures for decision making, policy selection, energy-based inference, and regularized optimization require evaluating either the minimum of a family of candidate costs or a softened variant of the corresponding argmin mechanism. However, the minimum operator is non-differentiable, which makes its direct integration into gradient-based pipelines difficult and often numerically

inconvenient [1–4]. For this reason, smooth surrogates are routinely introduced in order to retain tractability while preserving a meaningful approximation of the underlying optimization structure.

Among such surrogates, the entropic soft-min, or log-sum-exp approximation, is one of the most widely used constructions. Given a finite or measurable family of candidate costs, the entropic soft-min replaces the exact minimum by an exponentially weighted aggregation controlled by an inverse temperature parameter. This parameter governs the trade-off between sharpness and smoothness: as the inverse temperature increases, the relaxed operator approaches the exact minimum, while for smaller values the operator remains smoother and distributes mass over multiple candidates [2,5–8]. Variants of this mechanism arise throughout convex optimization, probabilistic inference, entropy-regularized learning, reinforcement learning, and receding-horizon control [2,6,9–11].

Despite its ubiquity, the selection of the inverse temperature parameter remains largely heuristic. In practice, one often fixes a single global value, chooses it by hand, or relies on deterministic worst-case estimates that depend only on the cardinality of the candidate set. Such strategies may be acceptable in homogeneous settings, but they become inadequate when the geometry of the candidate cost vector varies substantially across instances [12–16]. In particular, the approximation error of the entropic soft-min is not governed solely by the number of candidates. It also depends on how the near-optimal candidates are arranged around the minimum, that is, on the geometry of the excess-cost vector. Ignoring this geometry may lead either to inverse temperatures that are too small to control the approximation error, or to values that are unnecessarily conservative and destroy the intended smoothing effect [1,2,12,14].

This issue is especially relevant in machine learning and reinforcement learning applications, where candidate scores often exhibit substantial heterogeneity across contexts, states, or trajectories. In a contextual decision problem, some instances may display a sharply isolated optimum, whereas others may contain a broad near-optimal set. In tabular or approximate reinforcement learning, the local geometry of action-value estimates may vary significantly from one state to another, which directly affects the sharpness required for Gibbs-type action selection. Similar phenomena also arise in control and robotics, where soft-min aggregation is used to combine candidate controls or trajectories in the presence of prediction uncertainty and competing objectives [12–14,17,18]. Recent work on entropy-regularized softmax policy gradients further highlights the practical importance of temperature control in reinforcement learning pipelines [19]. In this study, however, we focus on operator-level calibration independently of the specific learning algorithm in which the soft-min is embedded.

The present work studies the calibration of the inverse temperature parameter from a geometry-aware perspective. Rather than viewing the entropic soft-min merely as a smooth approximation whose error decays with λ, we analyze how this error depends on structural descriptors of the candidate cost vector, such as the top-two gap and the size of the near-optimal set. This viewpoint shows that the effective sharpness required by the entropic relaxation depends on the geometry of the near-minimizers, and not only on support size.

To turn this observation into a practical selection rule, we adopt a conformal calibration framework. Conformal methods provide finite-sample, distribution-free validity under exchangeability, and are therefore well suited to parameter calibration problems in which one seeks guaranteed control of an error functional in future instances [20–24]. In the present setting, we treat the relaxation error of the entropic soft-min as the calibration score and construct a rule that selects the smallest inverse temperature ensuring that a prescribed upper quantile of the approximation error remains below a target tolerance. This yields

https://doi.org/10.3390/electronics15081704

a principled alternative to mean-based rules and worst-case deterministic bounds. This perspective is also consistent with recent developments in conformal calibration and risk control, including decision-aware conformal methods, end-to-end conformal risk training, and policy-level conformal regulation in sequential settings [25–27].

Beyond validity, the proposed calibration rule is adaptive to the geometry distribution represented in the calibration population. The method is geometry-aware at the population level. Broad near-optimal sets lead to smaller inverse temperatures, while isolated minima lead to larger values. This observation connects conformal quantile calibration with heterogeneous candidate-set geometry. It provides a principled mechanism for regime-adaptive soft-min tuning, formalized in Section 5 [28–30].

The paper develops this program in both theoretical and numerical terms. First, we introduce a geometry-aware formulation of the entropic soft-min and establish the basic invariance and structural properties of the associated relaxation error. We then derive geometry-dependent approximation bounds that make explicit the role of the near-optimal set. On this basis, we formulate a conformal calibration rule for the inverse temperature parameter and prove a finite-sample marginal validity guarantee under exchangeability. Finally, we evaluate the resulting method in three numerical settings: a geometry-controlled benchmark for excess-cost vectors, a conformal calibration benchmark comparing several inverse-temperature selection rules, and a tabular reinforcement learning experiment in which the local geometry of action-value vectors determines the effective softmax sharpness.

The preceding discussion highlights two complementary difficulties in inverse-temperature selection for entropic soft-min operators. Global heuristics and worst-case bounds do not reflect the geometry of the candidate cost vector, while practical tuning rules typically do not provide finite-sample guarantees of the resulting approximation error for future instances. The present work addresses these two points jointly by studying how the approximation error depends on the near-optimal structure of the candidate set and by using this perspective to construct a calibration rule with finite-sample operator-level validity.

The main contributions of the paper can be summarized as follows. First, a geometry-aware analysis of the entropic soft-min approximation error is developed, showing that the error depends on structural descriptors of the excess-cost vector and not merely on support size. Second, a conformal calibration rule is introduced that selects the smallest inverse temperature ensuring tolerance control of an upper quantile of the relaxation error with finite-sample distribution-free validity. Third, the resulting framework provides a principled mechanism for translating geometric heterogeneity of candidate cost vectors into calibrated soft-min sharpness. Fourth, the numerical experiments illustrate that this perspective remains meaningful in contextual decision and reinforcement learning-inspired settings, where Gibbs policies are induced from estimated action-value geometry.

The remainder of the paper is organized as follows. Section 2 introduces the geometry-aware entropic soft-min formulation and the associated geometric descriptors. Section 3 studies the geometry of the relaxation error. Section 4 derives geometry-dependent approximation bounds. Section 5 presents the conformal calibration rule and its validity guarantee. Section 6 reports the numerical experiments. Finally, Section 7 summarizes the main conclusions and discusses possible extensions.

2. Preliminaries and Geometry-Aware Entropic Relaxation

This section introduces the mathematical framework used throughout the paper and recalls the definition of the entropic soft-min operator. We also introduce the geometric descriptors used in the subsequent analysis.

Let $(\Phi, \mathcal{F}, \mu)$ be a measure space and let

$$\rho : \Phi \to \mathbb{R}$$

be a measurable cost function. Let $w : \Phi \to [0, \infty)$ be a probability density with respect to μ satisfying

$$\int_{\Phi} w(\phi)\, d\mu(\phi) = 1. \tag{1}$$

Let W denote the probability measure induced by w with respect to μ, namely

$$W(d\phi) = w(\phi)\, d\mu(\phi).$$

Then (1) implies that W is a probability measure on $(\Phi, \mathcal{F})$.

For $\lambda > 0$, define the partition function

$$Z_\lambda(\rho) = \int_{\Phi} e^{-\lambda \rho(\phi)}\, W(d\phi), \tag{2}$$

whenever the integral is finite. A sufficient condition for $0 < Z_\lambda(\rho) < \infty$ is that ρ is bounded from below [9,31]. In particular, if $\rho(\phi) \geq m$ for some $m \in \mathbb{R}$ and W is a probability measure, then

$$Z_\lambda(\rho) \leq e^{-\lambda m} < \infty.$$

The entropic soft-min (or log-sum-exp) operator is widely used in optimization and machine learning [2]. It is defined as follows.

Definition 1. *For $\lambda > 0$ and whenever $0 < Z_\lambda(\rho) < \infty$, the entropic soft-min relaxation of ρ is defined by*

$$D_\lambda(\rho) = -\frac{1}{\lambda} \log\left(\int_{\Phi} e^{-\lambda \rho(\phi)}\, W(d\phi) \right). \tag{3}$$

Here and throughout the paper, log *denotes the natural logarithm.*

The operator D_λ provides a smooth approximation of the infimum of ρ. As λ increases, the exponential weighting concentrates around low-cost regions, and the relaxation approaches the minimum value of the cost function.

Whenever $0 < Z_\lambda(\rho) < \infty$, the associated Gibbs probability measure [31] is defined by

$$\Pi_\lambda(d\phi) = \frac{e^{-\lambda \rho(\phi)}}{Z_\lambda(\rho)}\, W(d\phi). \tag{4}$$

Using (2), the entropic relaxation can be written equivalently as

$$D_\lambda(\rho) = -\frac{1}{\lambda} \log Z_\lambda(\rho). \tag{5}$$

While the general measure-theoretic formulation is useful for establishing structural properties, the subsequent analysis focuses on the finite-domain case, where the geometry of the candidate cost vector can be characterized explicitly.

In many optimization, machine learning, and inference problems, decisions are taken over finite candidate sets [2,9,10]. We therefore specialize the notation to the finite-domain setting.

Assume that

$$\Phi = \{1, \ldots, N\}, \qquad W(\{i\}) = w_i, \qquad w_i > 0, \qquad \sum_{i=1}^{N} w_i = 1.$$

In this finite-domain setting, the infimum is always attained, and therefore

$$\inf_{\phi\in\Phi}\rho(\phi)=\min_{1\le i\le N}\rho_i.$$

Let

$$\rho=(\rho_1,\dots,\rho_N)\in\mathbb{R}^N$$

denote the corresponding cost vector. Then (3) becomes

$$D_\lambda(\rho)=-\frac{1}{\lambda}\log\left(\sum_{i=1}^{N}w_i e^{-\lambda\rho_i}\right). \tag{6}$$

The uniform-weight case

$$w_i=\frac{1}{N},\qquad i=1,\dots,N,$$

plays a particularly important role and yields

$$D_\lambda(\rho)=-\frac{1}{\lambda}\log\left(\frac{1}{N}\sum_{i=1}^{N}e^{-\lambda\rho_i}\right). \tag{7}$$

The following result records a basic property of the entropic relaxation.

Proposition 1. *For every $\lambda>0$ such that $0<Z_\lambda(\rho)<\infty$,*

$$D_\lambda(\rho)\ge\inf_{\phi\in\Phi}\rho(\phi). \tag{8}$$

In the finite-domain case,

$$D_\lambda(\rho)\ge\min_{1\le i\le N}\rho_i. \tag{9}$$

Proof. Let

$$m=\inf_{\phi\in\Phi}\rho(\phi).$$

Then $\rho(\phi)\ge m$ for all $\phi\in\Phi$, hence

$$e^{-\lambda\rho(\phi)}\le e^{-\lambda m}.$$

Integrating with respect to $W(d\phi)$ gives

$$Z_\lambda(\rho)\le e^{-\lambda m}.$$

Substituting this bound into (5) yields

$$D_\lambda(\rho)=-\frac{1}{\lambda}\log Z_\lambda(\rho)\ge-\frac{1}{\lambda}\log(e^{-\lambda m})=m.$$

This proves (8). The finite-domain statement follows immediately. □

Since the entropic relaxation always overestimates the exact minimum value, it is natural to quantify the approximation quality through the associated relaxation error.

Definition 2. *For $\lambda>0$, define the relaxation error by*

$$\mathcal{E}_\lambda(\rho)=D_\lambda(\rho)-\inf_{\phi\in\Phi}\rho(\phi). \tag{10}$$

https://doi.org/10.3390/electronics15081704

In the finite-domain case,

$$\mathcal{E}_\lambda(\rho) = D_\lambda(\rho) - \min_{1\le i\le N} \rho_i. \tag{11}$$

Throughout the paper, $D_\lambda(\rho)$ denotes the entropic soft-min value, while $\mathcal{E}_\lambda(\rho)$ denotes the corresponding relaxation error. This notation will be used consistently in both the general and finite-domain settings.

By Proposition 1, one has

$$\mathcal{E}_\lambda(\rho) \ge 0.$$

The finite-domain formulation reveals that the approximation error depends only on the geometry of the cost vector relative to its minimum. This motivates the following centered representation.

Let

$$m(\rho) := \min_{1\le i\le N} \rho_i \tag{12}$$

and define the excess-cost vector

$$\delta(\rho) = \big(\delta_1(\rho), \dots, \delta_N(\rho)\big), \qquad \delta_i(\rho) := \rho_i - m(\rho) \ge 0. \tag{13}$$

Then at least one entry of $\delta(\rho)$ is zero.

The next identity shows that the relaxation error depends only on the excess-cost vector.

Proposition 2. *In the finite-domain setting, let*

$$m(\rho) = \min_{1\le i\le N} \rho_i$$

as defined in (12)*. Then*

$$D_\lambda(\rho) = m(\rho) - \frac{1}{\lambda} \log\left(\sum_{i=1}^{N} w_i e^{-\lambda \delta_i(\rho)} \right), \tag{14}$$

and therefore

$$\mathcal{E}_\lambda(\rho) = -\frac{1}{\lambda} \log\left(\sum_{i=1}^{N} w_i e^{-\lambda \delta_i(\rho)} \right). \tag{15}$$

In particular, in the uniform-weight case,

$$\mathcal{E}_\lambda(\rho) = -\frac{1}{\lambda} \log\left(\frac{1}{N} \sum_{i=1}^{N} e^{-\lambda \delta_i(\rho)} \right). \tag{16}$$

Proof. By definition,

$$\rho_i = m(\rho) + \delta_i(\rho), \qquad i = 1, \dots, N.$$

Substituting this into (6) gives

$$D_\lambda(\rho) = -\frac{1}{\lambda} \log\left(\sum_{i=1}^{N} w_i e^{-\lambda (m(\rho) + \delta_i(\rho))} \right).$$

Factoring out $e^{-\lambda m(\rho)}$ yields

$$D_\lambda(\rho) = -\frac{1}{\lambda} \log\left(e^{-\lambda m(\rho)} \sum_{i=1}^{N} w_i e^{-\lambda \delta_i(\rho)} \right),$$

which implies (14). Subtracting $m(\rho)$ proves (15), and (16) follows from $w_i = 1/N$. □

https://doi.org/10.3390/electronics15081704

Proposition 2 shows that the approximation problem is translation-invariant: adding a common constant to all candidate costs shifts the relaxed value by the same amount, while leaving the relaxation error unchanged.

Proposition 3. *Let* $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^N$ *and let* $c \in \mathbb{R}$. *Then, for every* $\lambda > 0$,

$$D_\lambda(\rho + c\mathbf{1}) = D_\lambda(\rho) + c \tag{17}$$

and

$$\mathcal{E}_\lambda(\rho + c\mathbf{1}) = \mathcal{E}_\lambda(\rho). \tag{18}$$

Proof. Recall that W is a probability measure, i.e., $W(\Phi) = 1$. Using (6),

$$D_\lambda(\rho + c\mathbf{1}) = -\frac{1}{\lambda} \log\left(\sum_{i=1}^{N} w_i e^{-\lambda(\rho_i + c)} \right) = -\frac{1}{\lambda} \log\left(e^{-\lambda c} \sum_{i=1}^{N} w_i e^{-\lambda \rho_i} \right),$$

which gives

$$D_\lambda(\rho + c\mathbf{1}) = c + D_\lambda(\rho).$$

Also,

$$\min_i(\rho_i + c) = \min_i \rho_i + c.$$

Subtracting the minima on both sides yields (18). □

A second invariance property concerns relabeling of the candidates.

Proposition 4. *Assume the uniform-weight finite-domain setting, i.e.,* $w_i = 1/N$ *for all* i. *Let* π *be any permutation of* $\{1, \ldots, N\}$ *and define*

$$\rho^\pi = (\rho_{\pi(1)}, \ldots, \rho_{\pi(N)}).$$

Then, for every $\lambda > 0$,

$$D_\lambda(\rho^\pi) = D_\lambda(\rho) \tag{19}$$

and

$$\mathcal{E}_\lambda(\rho^\pi) = \mathcal{E}_\lambda(\rho). \tag{20}$$

Proof. Under uniform weights,

$$D_\lambda(\rho^\pi) = -\frac{1}{\lambda} \log\left(\frac{1}{N} \sum_{i=1}^{N} e^{-\lambda \rho_{\pi(i)}} \right).$$

Since permutation does not change the value of the sum,

$$\sum_{i=1}^{N} e^{-\lambda \rho_{\pi(i)}} = \sum_{i=1}^{N} e^{-\lambda \rho_i},$$

and therefore (19) holds. Also,

$$\min_i \rho_{\pi(i)} = \min_i \rho_i,$$

which implies (20). □

The invariance property relies on the uniform-weight assumption. For general weights, the value of $D_\lambda(\rho)$ depends on the pairing between weights and costs, and therefore permutation invariance does not hold unless the weights are permuted accordingly.

https://doi.org/10.3390/electronics15081704

The previous invariance properties show that the approximation error depends on the geometry of the excess costs rather than on the raw cost vector. We now introduce the corresponding geometric descriptors.

Let

$$0 = \delta_{(1)}(\rho) \le \delta_{(2)}(\rho) \le \cdots \le \delta_{(N)}(\rho) \tag{21}$$

denote the ordered values of the excess-cost vector (13).

Definition 3. *The* top-two gap *of the cost vector ρ is defined by*

$$\Delta_{(2)}(\rho) := \delta_{(2)}(\rho) = \rho_{(2)} - \rho_{(1)}, \tag{22}$$

where $\rho_{(1)} \le \rho_{(2)} \le \cdots \le \rho_{(N)}$ are the ordered costs.

For $\varepsilon > 0$, the ε-near-optimal mass *of ρ is defined by*

$$N_\varepsilon(\rho) := \#\big\{i \in \{1, \dots, N\} : \delta_i(\rho) \le \varepsilon\big\}. \tag{23}$$

The descriptor $\Delta_{(2)}(\rho)$ measures the separation between the optimal candidate and its nearest competitor, while $N_\varepsilon(\rho)$ measures the size of the near-optimal set at tolerance level ε. These quantities will play a central role in the geometry-dependent analysis of the approximation error developed in the next sections.

In particular, the geometry of the near-optimal set influences the concentration pattern of the Gibbs weights [10,32]

$$p_i(\lambda; \rho) = \frac{w_i e^{-\lambda \rho_i}}{\sum_{j=1}^N w_j e^{-\lambda \rho_j}}, \qquad i = 1, \dots, N. \tag{24}$$

In the uniform-weight case, (24) simplifies to

$$p_i(\lambda; \rho) = \frac{e^{-\lambda \rho_i}}{\sum_{j=1}^N e^{-\lambda \rho_j}}. \tag{25}$$

As λ increases, these weights become more concentrated on the indices with smallest cost. However, the rate at which this concentration occurs depends on the geometry encoded by (21), (22) and (23).

Finally, for later use, we record the standard variational representation of the entropic relaxation.

Proposition 5. *For every $\lambda > 0$, such that $0 < Z_\lambda(\rho) < \infty$,*

$$D_\lambda(\rho) = \inf_{Q \ll W} \left(\mathbb{E}_Q[\rho] + \frac{1}{\lambda} \mathrm{KL}(Q \| W) \right), \tag{26}$$

where W is a probability measure and the infimum is taken over probability measures Q on $(\Phi, \mathcal{F})$ that are absolutely continuous with respect to W, so that the Kullback–Leibler divergence $\mathrm{KL}(Q\|W)$ is well defined.

This representation follows from the variational characterization of the log-partition function and the Gibbs variational principle [9,31,33–36]. Indeed, for any probability measure $Q \ll W$, one may write

$$Z_\lambda(\rho) = \mathbb{E}_Q\left[\exp\left(-\lambda \rho - \log \frac{dQ}{dW} \right) \right],$$

and Jensen's inequality yields

$$D_\lambda(\rho) \le \mathbb{E}_Q[\rho] + \frac{1}{\lambda}\mathrm{KL}(Q\|W).$$

Taking the infimum over $Q \ll W$ and evaluating at the Gibbs measure Π_λ gives (26). The representation shows that the entropic soft-min balances expected cost and divergence from the reference measure W.

Remark 1. *By Proposition 2, the finite-domain relaxation error is given by* (15) *for arbitrary weights. Consequently, the score construction used in Section 5 can be formulated in the same way under a nonuniform weight profile. However, in the weighted case the approximation error depends jointly on the excess-cost vector and on the weights. In particular, the permutation invariance property established below for the uniform-weight setting is no longer available unless costs and weights are relabeled jointly, and descriptors based only on cardinality must be replaced by weighted analogues. A natural example is the weighted near-optimal mass*

$$M_\varepsilon^w(\rho) := \sum_{i:\delta_i(\rho)\le\varepsilon} w_i,$$

which replaces the cardinality-based quantity $N_\varepsilon(\rho)$ in the nonuniform setting. A full geometry-dependent extension of the bounds and proofs to arbitrary weights is left for future work. The results of Sections 3 and 4 are therefore developed under the assumption of uniform weights.

3. Geometry of the Relaxation Error

We study the dependence of the approximation accuracy of the entropic soft-min operator on the geometry of the candidate cost vector. In this section, we consider the finite-domain setting

$$\Phi = \{1, \dots, N\},$$

and we consider the uniform-weight formulation

$$D_\lambda(\rho) = -\frac{1}{\lambda}\log\left(\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda\rho_i}\right), \tag{27}$$

which is the form commonly used in optimization, machine learning, and reinforcement learning applications.

Let

$$m(\rho) = \min_{1\le i\le N} \rho_i$$

and define the excess-cost vector

$$\delta_i(\rho) = \rho_i - m(\rho).$$

Using Proposition 2, the relaxation error can be written as

$$\mathcal{E}_\lambda(\rho) = -\frac{1}{\lambda}\log\left(\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda\delta_i(\rho)}\right). \tag{28}$$

Equation (28) shows that the approximation error depends only on the geometry of the excess-cost vector $\delta(\rho)$.

The main intuition is that the relaxation error is governed not only by the number of candidates, but by how the near-optimal candidates are arranged around the minimum. When several competitors remain close to the optimum, the exponential averaging retains

https://doi.org/10.3390/electronics15081704

more mass near the minimum and the relaxation behaves differently than in a sharply isolated-minimum regime. The results of this section make this dependence precise through the structural properties of the error.

3.1. Basic Structural Properties

The relaxation error is closely related to log-partition functions and Gibbs measures [31,32]. We now establish the basic properties of this quantity.

Lemma 1. *For any cost vector $\rho \in \mathbb{R}^N$, the mapping*

$$\lambda \mapsto \mathcal{E}_\lambda(\rho)$$

is nonincreasing on $(0, \infty)$.

Proof. Let

$$Z_\lambda(\rho) = \frac{1}{N}\sum_{i=1}^{N} e^{-\lambda\rho_i}, \qquad D_\lambda(\rho) = -\frac{1}{\lambda}\log Z_\lambda(\rho).$$

Differentiating with respect to λ and using the Gibbs distribution

$$p_i(\lambda) = \frac{e^{-\lambda\rho_i}}{\sum_{j=1}^{N} e^{-\lambda\rho_j}},$$

one obtains

$$\frac{d}{d\lambda}D_\lambda(\rho) = -\frac{1}{\lambda^2}\mathrm{KL}(p(\lambda)\|\mathrm{Unif}),$$

where Unif denotes the uniform distribution on $\{1, \ldots, N\}$.

Since the Kullback–Leibler divergence is nonnegative, it follows that

$$\frac{d}{d\lambda}D_\lambda(\rho) \leq 0.$$

The result follows because $\min_i \rho_i$ does not depend on λ. □

Lemma 1 shows that the approximation error is nonincreasing in λ. The following result provides a bound.

Proposition 6. *For every $\rho \in \mathbb{R}^N$ and $\lambda > 0$,*

$$0 \leq \mathcal{E}_\lambda(\rho) \leq \frac{\log N}{\lambda}. \tag{29}$$

Proof. Let $m = \min_i \rho_i$. Then

$$\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda\rho_i} = e^{-\lambda m}\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda(\rho_i - m)}.$$

Since $\rho_i - m \geq 0$, one has

$$\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda(\rho_i - m)} \leq 1.$$

Substituting into (27) gives

$$D_\lambda(\rho) \geq m.$$

https://doi.org/10.3390/electronics15081704

On the other hand,

$$\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda\rho_i} \geq \frac{1}{N}e^{-\lambda m}.$$

Taking logarithms yields

$$D_\lambda(\rho) \leq m + \frac{\log N}{\lambda}.$$

Subtracting m proves (29). □

The bound (29) depends only on N and is attained when the minimum is unique and the remaining costs are large. However, this bound does not reflect the geometry of the cost vector and is often overly conservative in practice.

3.2. Geometry-Dependent Representation

We now characterize how the relaxation error depends on the geometry of the near-optimal candidates.

Let

$$0 = \delta_{(1)} \leq \delta_{(2)} \leq \cdots \leq \delta_{(N)}$$

denote the ordered excess costs defined in (21).

Proposition 7. *The relaxation error satisfies*

$$\mathcal{E}_\lambda(\rho) = -\frac{1}{\lambda}\log\left(\frac{1}{N}\left(1 + \sum_{i=2}^{N} e^{-\lambda\delta_{(i)}}\right)\right). \tag{30}$$

Proof. By definition,

$$\delta_{(1)} = 0.$$

Substituting into (28) yields

$$\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda\delta_i} = \frac{1}{N}\left(e^{-\lambda\delta_{(1)}} + \sum_{i=2}^{N} e^{-\lambda\delta_{(i)}}\right).$$

Since $\delta_{(1)} = 0$, one obtains

$$\frac{1}{N}\left(1 + \sum_{i=2}^{N} e^{-\lambda\delta_{(i)}}\right),$$

which proves the claim. □

The quantity $\Delta_{(2)}(\rho) = \rho_{(2)} - \rho_{(1)} = \delta_{(2)}(\rho)$ represents the top-two gap, i.e., the separation between the optimal candidate and its nearest competitor. Expression (30) shows that the relaxation error depends only on the distribution of the excess costs above the minimum. The remaining terms decay exponentially in λ and in $\delta_{(i)}(\rho)$, so that higher-cost candidates rapidly become negligible. Thus, the approximation error depends only on the geometry of the excess-cost vector.

3.3. Dependence on the Near-Optimal Geometry

We now relate the approximation error to the geometry of the near-optimal set. Let $\varepsilon > 0$ and recall the definition of the near-optimal mass

$$N_\varepsilon(\rho) = \#\{i : \delta_i(\rho) \leq \varepsilon\}.$$

https://doi.org/10.3390/electronics15081704

Proposition 8. *For any $\varepsilon > 0$,*

$$\mathcal{E}_\lambda(\rho) \le \frac{1}{\lambda} \log\left(\frac{N}{N_\varepsilon(\rho)}\right) + \varepsilon. \tag{31}$$

Proof. Split the sum in (28) as

$$\sum_{i=1}^{N} e^{-\lambda\delta_i} = \sum_{\delta_i \le \varepsilon} e^{-\lambda\delta_i} + \sum_{\delta_i > \varepsilon} e^{-\lambda\delta_i}.$$

Since $\delta_i \le \varepsilon$ for the first group,

$$e^{-\lambda\delta_i} \ge e^{-\lambda\varepsilon}.$$

Therefore

$$\sum_{i=1}^{N} e^{-\lambda\delta_i} \ge N_\varepsilon(\rho) e^{-\lambda\varepsilon}.$$

Substituting into (28) gives

$$\mathcal{E}_\lambda(\rho) \le -\frac{1}{\lambda} \log\left(\frac{N_\varepsilon(\rho)}{N} e^{-\lambda\varepsilon}\right),$$

which simplifies to (31). □

Proposition 8 shows that the approximation accuracy depends on the number of near-optimal candidates. In particular, several near-tied candidates reduce the relaxation error relative to the worst-case bound. This observation is consistent with previous analyses of soft-min approximations [13,18].

3.4. Asymptotic Behavior

We now study the asymptotic behavior of the relaxation error as the inverse temperature tends to infinity.

Proposition 9. *Let*

$$k = \#\{i : \rho_i = m(\rho)\}$$

denote the multiplicity of the minimum. Then

$$\mathcal{E}_\lambda(\rho) = \frac{1}{\lambda} \log(N/k) + o\left(\frac{1}{\lambda}\right) \quad \text{as } \lambda \to \infty. \tag{32}$$

Proof. Let $m = \min_i \rho_i$. Then

$$\sum_{i=1}^{N} e^{-\lambda\rho_i} = k e^{-\lambda m} + \sum_{\rho_i > m} e^{-\lambda\rho_i}.$$

Define

$$\delta = \min_{\rho_i > m} (\rho_i - m) > 0.$$

Then

$$\sum_{\rho_i > m} e^{-\lambda\rho_i} = O(e^{-\lambda(m+\delta)}).$$

Hence

$$\frac{1}{N} \sum_{i=1}^{N} e^{-\lambda\rho_i} = \frac{k}{N} e^{-\lambda m}(1 + o(1)).$$

https://doi.org/10.3390/electronics15081704

Substituting into (27) yields the result. □

Proposition 9 shows that the asymptotic approximation error is determined by the multiplicity of the minimizers. In particular, when the minimum is unique ($k = 1$), the leading term reduces to $\frac{\log N}{\lambda}$, which coincides with the worst-case bound (29). When several candidates attain the minimum ($k > 1$), the effective support size reduces from N to N/k, and the approximation error improves accordingly. This asymptotic regime is consistent with classical Laplace-type arguments for exponentially weighted sums and integrals [34,37–39].

Hence, the approximation error is determined by the geometry of the excess-cost vector. In particular, near-tied candidates lead to smaller errors than those predicted by the worst-case bound.

Taken together, these results show that the relaxation error is controlled by a small set of geometric features, in particular the top-two gap, the near-optimal mass, and the multiplicity of the minimum. This provides the structural motivation for the more explicit geometry-dependent bounds developed in Section 5.

4. Geometry-Dependent Approximation Bounds

The structural properties established in Section 3 show that the approximation error of the entropic soft-min operator depends only on the geometry of the excess-cost vector introduced in (13). In this section we derive bounds that make this dependence explicit.

Throughout the section we consider the finite-domain formulation

$$D_\lambda(\rho) = -\frac{1}{\lambda} \log\left(\frac{1}{N} \sum_{i=1}^{N} e^{-\lambda \rho_i} \right), \tag{33}$$

and we denote by

$$\delta_i(\rho) = \rho_i - m(\rho), \qquad m(\rho) = \min_i \rho_i,$$

the excess-cost representation introduced in Section 2.

Using Proposition 2, the relaxation error can be written as

$$\mathcal{E}_\lambda(\rho) = -\frac{1}{\lambda} \log\left(\frac{1}{N} \sum_{i=1}^{N} e^{-\lambda \delta_i(\rho)} \right). \tag{34}$$

This representation makes it possible to derive approximation bounds that depend on the geometry of the excess-cost vector.

4.1. Classical Worst-Case Bound

We recall the classical worst-case bound [1,2,5], which depends only on the number of candidates and follows directly from Proposition 6. For every $\rho \in \mathbb{R}^N$ and every $\lambda > 0$,

$$\mathcal{E}_\lambda(\rho) \leq \frac{\log N}{\lambda}. \tag{35}$$

The bound (35) is distribution-free, but ignores the geometry of the candidate cost vector and may be conservative. The upper bound $\frac{\log N}{\lambda}$ is tight in the limit where the minimum is unique and the remaining costs diverge to infinity, so that only one term contributes to the exponential sum.

4.2. Bounds Based on the Near-Optimal Set

We now derive bounds that depend explicitly on the geometry of the near-optimal candidates.

https://doi.org/10.3390/electronics15081704

Let

$$N_\varepsilon(\rho) = \#\{i : \delta_i(\rho) \le \varepsilon\}$$

denote the ε-near-optimal mass introduced in Definition 3.

Theorem 1. *For every $\lambda > 0$ and $\varepsilon > 0$,*

$$\mathcal{E}_\lambda(\rho) \le \frac{1}{\lambda} \log\left(\frac{N}{N_\varepsilon(\rho)}\right) + \varepsilon. \tag{36}$$

Proof. Using the representation (34), write

$$\sum_{i=1}^{N} e^{-\lambda\delta_i} = \sum_{\delta_i \le \varepsilon} e^{-\lambda\delta_i} + \sum_{\delta_i > \varepsilon} e^{-\lambda\delta_i}.$$

For the first group we have $\delta_i \le \varepsilon$, hence

$$e^{-\lambda\delta_i} \ge e^{-\lambda\varepsilon}.$$

Therefore

$$\sum_{i=1}^{N} e^{-\lambda\delta_i} \ge N_\varepsilon(\rho) e^{-\lambda\varepsilon}.$$

Substituting into (34) gives

$$\mathcal{E}_\lambda(\rho) \le -\frac{1}{\lambda} \log\left(\frac{N_\varepsilon(\rho)}{N} e^{-\lambda\varepsilon}\right),$$

which simplifies to (36). □

The bound (36) shows that the approximation accuracy depends on the number of candidates whose costs lie close to the minimum.

When several candidates have costs close to the minimum, the relaxation error decreases.

4.3. Bounds Based on the Top-Two Gap

Another useful geometric descriptor is the gap between the two smallest costs.

Let

$$\Delta_{(2)}(\rho) = \rho_{(2)} - \rho_{(1)}, \tag{37}$$

where $\rho_{(1)} \le \rho_{(2)} \le \cdots \le \rho_{(N)}$ denote the ordered costs.

Proposition 10. *For every $\lambda > 0$, the relaxation error satisfies*

$$\frac{1}{\lambda}\left[\log N - \log\left(1 + (N-1)e^{-\lambda\Delta_{(2)}(\rho)}\right)\right] \le \mathcal{E}_\lambda(\rho) \le \frac{1}{\lambda}\left[\log N - \log\left(1 + e^{-\lambda\Delta_{(2)}(\rho)}\right)\right]. \tag{38}$$

Proof. Using the ordered excess-cost representation (30), one has

$$\mathcal{E}_\lambda(\rho) = \frac{1}{\lambda}\left[\log N - \log\left(1 + \sum_{i=2}^{N} e^{-\lambda\delta_{(i)}}\right)\right].$$

Since $\delta_{(i)} \ge \Delta_{(2)}(\rho)$ for all $i \ge 2$, it follows that

$$\sum_{i=2}^{N} e^{-\lambda\delta_{(i)}} \le (N-1)e^{-\lambda\Delta_{(2)}(\rho)}.$$

https://doi.org/10.3390/electronics15081704

This inequality follows from the fact that $\delta_{(i)} \geq \Delta_{(2)}(\rho)$ for all $i \geq 2$, so each exponential term is bounded above by $e^{-\lambda\Delta_{(2)}(\rho)}$. Substituting this bound into the logarithm yields

$$\mathcal{E}_\lambda(\rho) \geq \frac{1}{\lambda}\Big[\log N - \log\Big(1 + (N-1)e^{-\lambda\Delta_{(2)}(\rho)}\Big)\Big].$$

On the other hand, since $\delta_{(2)} = \Delta_{(2)}(\rho)$ and all summands are nonnegative,

$$\sum_{i=2}^{N} e^{-\lambda\delta_{(i)}} \geq e^{-\lambda\Delta_{(2)}(\rho)}.$$

Substituting again gives

$$\mathcal{E}_\lambda(\rho) \leq \frac{1}{\lambda}\Big[\log N - \log\Big(1 + e^{-\lambda\Delta_{(2)}(\rho)}\Big)\Big].$$

□

The bounds in (38) make explicit that the top-two gap $\Delta_{(2)}(\rho)$ is the primary geometric quantity controlling the contribution of nonminimal candidates. As $\Delta_{(2)}(\rho)$ increases, the exponential terms $e^{-\lambda\Delta_{(2)}(\rho)}$ decay rapidly, so that the contribution of the remaining candidates becomes negligible and the relaxation error approaches the asymptotic regime determined by the multiplicity of the minimum. In particular, large gaps lead to small approximation errors, even for moderate values of λ.

4.4. Asymptotic Geometry of the Approximation Error

We study the asymptotic behavior of the relaxation error. Let

$$k = \#\{i : \rho_i = m(\rho)\}$$

denote the multiplicity of the minimum.

Theorem 2. *As* $\lambda \to \infty$,

$$\mathcal{E}_\lambda(\rho) = \frac{1}{\lambda}\log\left(\frac{N}{k}\right) + o\left(\frac{1}{\lambda}\right). \tag{39}$$

As $\lambda \to \infty$, only the candidates attaining the minimum contribute to the exponential sum. Consequently,

$$\mathcal{E}_\lambda(\rho) \approx \frac{\log N - \log k}{\lambda},$$

where k is the multiplicity of the minimum. In particular, a unique minimizer ($k = 1$) yields the worst-case rate $\frac{\log N}{\lambda}$, whereas multiple minimizers reduce the effective support to k, leading to $\frac{\log(N/k)}{\lambda}$. This emphasizes that the approximation error is governed by the geometry of the near-optimal set rather than by the full support size [34,37–39].

5. Conformal Calibration of the Inverse Temperature

The bounds derived in Section 4 characterize how the approximation error of the entropic soft-min operator depends on the inverse temperature parameter λ and on the geometry of the candidate cost vector. In practice, however, the geometry of future cost vectors is unknown. Consequently, selecting λ directly from these bounds is generally not possible [20,24,40].

The calibration procedure developed in this section relies on the exchangeability assumption standard in conformal inference. In particular, we assume that the calibration and test cost vectors are i.i.d. This assumption is essential for the finite-sample validity

https://doi.org/10.3390/electronics15081704

guarantee established below and does not cover settings with temporal dependence or distributional shifts.

5.1. Calibration Setting

Suppose that we observe a calibration sample of cost vectors

$$\rho^{(1)}, \dots, \rho^{(n)} \in \mathbb{R}^N,$$

and let $\rho^{(n+1)}$ denote a future test cost vector.

The theoretical guarantee established below assumes that

$$\rho^{(1)}, \dots, \rho^{(n+1)}$$

are exchangeable random vectors. This assumption is critical for the validity guarantee. It requires that the calibration instances and the future test instance are identically distributed and jointly exchangeable, which in particular excludes distributional shifts, temporal dependence, or adaptive data-generation mechanisms. In such cases, the finite-sample guarantee (45) may no longer hold.

In the present work, exchangeability should be interpreted as the standard i.i.d. assumption underlying conformal calibration, under which the theoretical coverage guarantee is valid.

Partial relaxations of exchangeability have been studied in related conformal settings. In particular, weighted conformal constructions have been developed for covariate shift, where validity is transferred to a target population through likelihood-ratio weighting [41]. Beyond exchangeability, recent results show that split conformal procedures may retain approximate validity under suitable non-exchangeable regimes, including settings with temporal or weak dependence [40,42,43]. In sequential settings, online conformal and conformal-martingale constructions provide alternative validity mechanisms adapted to data arriving over time [44,45]. These directions are relevant for dynamic decision problems, but their adaptation to the present operator-level calibration framework, and in particular to multi-step reinforcement learning settings with state dependence and policy-induced distribution shift, is beyond the scope of this paper.

For each cost vector ρ, recall the definition of the relaxation error

$$\mathcal{E}_\lambda(\rho) = D_\lambda(\rho) - \min_i \rho_i,$$

introduced in Definition 2.

For each calibration instance we evaluate the relaxation error as a function of the inverse temperature parameter.

Definition 4. *For each calibration instance $\rho^{(j)}$, define the score function*

$$S_j(\lambda) = \mathcal{E}_\lambda(\rho^{(j)}), \qquad j = 1, \dots, n. \tag{40}$$

The score $S_j(\lambda)$ measures the discrepancy between the entropic relaxation and the exact minimum for the j-th calibration instance.

We now introduce the instance-wise inverse temperature required to achieve a prescribed tolerance.

Definition 5. *Let $\tau > 0$ be a prescribed tolerance level. For a cost vector ρ, define*

$$\Lambda(\rho) := \inf\{\lambda > 0 : \mathcal{E}_\lambda(\rho) \le \tau\}, \tag{41}$$

https://doi.org/10.3390/electronics15081704

whenever the set on the right-hand side is nonempty.

If no such λ exists, we set $\Lambda(\rho) = +\infty$.

In the present setting, the monotonicity and continuity of $\mathcal{E}_\lambda(\rho)$ imply that the set is either empty or of the form $[\Lambda(\rho), \infty)$, and is nonempty whenever the tolerance τ exceeds the infimum of $\mathcal{E}_\lambda(\rho)$. Thus, whenever the set is nonempty, $\Lambda(\rho)$ is the smallest inverse temperature for which the relaxation error of ρ does not exceed τ.

By Lemma 1, the mapping

$$\lambda \mapsto \mathcal{E}_\lambda(\rho)$$

is nonincreasing on $(0, \infty)$. Moreover, from the representation (28), it follows that $\mathcal{E}_\lambda(\rho)$ is continuous in λ, since the function $\lambda \mapsto \sum_{i=1}^{N} e^{-\lambda\delta_i(\rho)}$ is continuous and strictly positive, and the logarithm and scaling by $1/\lambda$ preserve continuity on $(0, \infty)$.

Consequently, the set

$$\{\lambda > 0 : \mathcal{E}_\lambda(\rho) \leq \tau\}$$

is either empty or of the form $[\Lambda(\rho), \infty)$, and the infimum in (41) is attained whenever the set is nonempty. In particular, $\Lambda(\rho)$ is well defined as the smallest inverse temperature achieving tolerance.

5.2. Empirical Calibration Rule

For a fixed λ, we compute the calibration scores

$$S_1(\lambda), \ldots, S_n(\lambda).$$

Let

$$S_{(1)}(\lambda) \leq \cdots \leq S_{(n)}(\lambda)$$

denote the ordered values.

Definition 6. *For $\alpha \in (0, 1)$ such that $\alpha \geq 1/(n+1)$, define the empirical $(1-\alpha)$ quantile*

$$\widehat{q}_{1-\alpha}(\lambda) = S_{(k)}(\lambda), \qquad k = \lceil (1-\alpha)(n+1) \rceil. \tag{42}$$

The choice of the index $k = \lceil (1-\alpha)(n+1) \rceil$ follows the standard quantile selection rule in conformal calibration. In particular, the empirical quantile $\widehat{q}_{1-\alpha}(\lambda)$ is defined as an order statistic that upper bounds the $(1-\alpha)$-quantile of the score distribution based on a finite sample [20–22].

This specific choice introduces a finite-sample correction that ensures a conservative estimate of the target quantile, thereby providing a safety margin that guarantees the desired coverage level $1-\alpha$ under exchangeability.

The condition $\alpha \geq 1/(n+1)$ ensures that the index

$$k = \lceil (1-\alpha)(n+1) \rceil$$

satisfies $1 \leq k \leq n$, so that the empirical quantile is well defined. If $\alpha < 1/(n+1)$, then $k = n+1$, and no valid order statistic from the calibration sample can be assigned.

The next lemma shows that the empirical quantile inherits the same monotonicity in λ as the individual relaxation errors.

Lemma 2. *The function*

$$\lambda \mapsto \widehat{q}_{1-\alpha}(\lambda)$$

is nonincreasing.

https://doi.org/10.3390/electronics15081704

Proof. By Lemma 1, for each instance $\rho^{(j)}$ the mapping

$$\lambda \mapsto \mathcal{E}_\lambda(\rho^{(j)})$$

is nonincreasing. Therefore

$$S_j(\lambda_2) \le S_j(\lambda_1) \qquad \text{whenever } \lambda_2 > \lambda_1.$$

Thus the entire sample of scores decreases componentwise as λ increases. Consequently, their empirical quantile cannot increase. □

The monotonicity of $\widehat{q}_{1-\alpha}(\lambda)$ implies that the constraint

$$\widehat{q}_{1-\alpha}(\lambda) \le \tau$$

defines a threshold condition in λ.

Definition 7. *Define the calibrated inverse temperature*

$$\lambda^* = \inf\{\lambda > 0 : \widehat{q}_{1-\alpha}(\lambda) \le \tau\}. \tag{43}$$

Thus λ^* is the smallest inverse temperature for which the empirical upper quantile of the relaxation error does not exceed the tolerance.

5.3. Equivalent Order-Statistic Representation

The calibration rule admits a simple equivalent representation.

Lemma 3. *Let*

$$\Lambda_j = \Lambda(\rho^{(j)}), \qquad j = 1, \dots, n,$$

and let

$$\Lambda_{(1)} \le \cdots \le \Lambda_{(n)}$$

denote their ordered values.

Then

$$\lambda^* = \Lambda_{(k)}, \qquad k = \lceil (1-\alpha)(n+1) \rceil. \tag{44}$$

Proof. By Definition 5,

$$\Lambda_j \le \lambda \quad \Longleftrightarrow \quad \mathcal{E}_\lambda(\rho^{(j)}) \le \tau.$$

Thus the condition

$$\widehat{q}_{1-\alpha}(\lambda) \le \tau$$

is equivalent to requiring that at least k calibration instances satisfy

$$\mathcal{E}_\lambda(\rho^{(j)}) \le \tau.$$

Equivalently, at least k values Λ_j satisfy

$$\Lambda_j \le \lambda.$$

The smallest λ with this property is $\Lambda_{(k)}$. □

5.4. Finite-Sample Validity

We now establish the main guarantee of the calibration rule.

https://doi.org/10.3390/electronics15081704

Theorem 3. *Assume that the sequence*

$$\rho^{(1)}, \ldots, \rho^{(n+1)}$$

is exchangeable.

Let λ^* be defined by (43)*. Then*

$$\mathbb{P}\Big(\mathcal{E}_{\lambda^*}(\rho^{(n+1)}) \leq \tau\Big) \geq 1 - \alpha, \tag{45}$$

where the probability is taken over the joint distribution of the calibration sample and the test instance.

Proof. Let

$$\Lambda_i = \Lambda(\rho^{(i)}), \qquad i = 1, \ldots, n+1.$$

By the assumed exchangeability of $\rho^{(1)}, \ldots, \rho^{(n+1)}$, the random variables

$$\Lambda_1, \ldots, \Lambda_{n+1}$$

are also exchangeable.

By Lemma 3, the calibrated inverse temperature satisfies

$$\lambda^* = \Lambda_{(k)}.$$

By exchangeability, the test value Λ_{n+1} is equally likely to occupy any position among the $n+1$ exchangeable values, up to ties. Therefore,

$$\mathbb{P}\Big(\Lambda_{n+1} \leq \Lambda_{(k)}\Big) \geq \frac{k}{n+1} \geq 1 - \alpha.$$

Therefore

$$\mathbb{P}\Big(\Lambda_{n+1} \leq \Lambda_{(k)}\Big) \geq 1 - \alpha.$$

By Definition 5,

$$\Lambda_{n+1} \leq \lambda^* \quad \Longrightarrow \quad \mathcal{E}_{\lambda^*}(\rho^{(n+1)}) \leq \tau.$$

Combining these relations yields (45). □

The guarantee is marginal. The coverage level $1 - \alpha$ holds over the joint distribution of calibration and test samples and may vary for a fixed calibration set. The calibrated inverse temperature λ^* defines a nonparametric tolerance threshold [20,23,40].

5.5. Interpretation

Theorem 3 provides a distribution-free guarantee on the approximation accuracy of the entropic soft-min operator. The calibrated inverse temperature λ^* ensures that the relaxation error satisfies the tolerance constraint with probability at least $1 - \alpha$ for a future instance drawn from the same exchangeable population.

Importantly, the calibration rule automatically adapts to the geometry of the cost vectors represented in the calibration sample. As shown in Section 4, different geometries of the excess-cost vector may require different inverse temperatures in order to achieve the same tolerance level. The conformal calibration rule selects the smallest inverse temperature that satisfies the tolerance constraint at the prescribed coverage level for the calibration population.

Consequently, the method provides a principled alternative to global heuristic tuning rules and worst-case bounds that ignore the geometry of the candidate cost vectors.

https://doi.org/10.3390/electronics15081704

In practical terms, the method is most naturally suited to repeated decision problems in which a calibration sample can be collected from a population that is representative of future instances. Under this condition, the selected inverse temperature can be interpreted as a population-level operating point that controls the upper tail of the relaxation error at the prescribed confidence level. The same interpretation becomes less reliable when the calibration sample is not representative, for example under temporal dependence, distribution shift, or changes in the geometry distribution of the candidate sets. In addition, while the conformal score construction extends directly to nonuniform weights, the geometry-dependent analysis developed here is restricted to the uniform-weight case. These assumptions do not limit the internal validity of the present results, but they delimit the practical regime in which the current calibration rule can be expected to transfer without further adaptation.

6. Numerical Experiments

This section evaluates the proposed conformal calibration rule for the inverse temperature parameter through controlled numerical experiments, illustrating the geometry-dependent behavior of the entropic relaxation error, the empirical validity of the calibration rule, and its interaction with learning dynamics in reinforcement learning settings.

All simulations were implemented in MATLAB 2024b using fixed random seeds. Whenever several methods are compared on the same episode, they are evaluated under identical realizations of the environment and candidate sets in order to eliminate Monte Carlo variability from the comparison.

Unless otherwise stated, the finite-domain entropic relaxation

$$D_\lambda(\rho) = -\frac{1}{\lambda}\log\left(\frac{1}{N}\sum_{i=1}^{N} e^{-\lambda\rho_i}\right) \tag{46}$$

is used, and the relaxation error is computed as

$$\mathcal{E}_\lambda(\rho) = D_\lambda(\rho) - \min_i \rho_i.$$

The calibration procedure introduced in Section 5 is applied with target coverage level $1-\alpha = 0.9$ and tolerance $\tau = 0.1$.

Taken together, the numerical study combines controlled geometry ablations, sensitivity analyses with respect to the population mixture, and a computational-overhead evaluation, so that both the statistical and computational behavior of the proposed calibration rule can be assessed explicitly.

Throughout the experiments, the compared baselines are restricted to selectors that output a single inverse temperature for the same static operator-level calibration problem. By contrast, annealing schedules or learnable temperature mechanisms of the type used in entropy-regularized reinforcement learning algorithms are coupled to iterative training dynamics and to objectives different from tolerance control of the soft-min approximation. They are therefore not directly commensurate with the calibration problem studied in the present work.

6.1. Experiment 1: Geometry-Controlled Cost Vectors

The first experiment studies the behavior of the entropic relaxation error under controlled candidate-set geometries. The objective is to illustrate how the geometry of the excess-cost vector influences the approximation accuracy of the soft-min operator.

For each trial, a cost vector

$$\rho = (\rho_1, \ldots, \rho_N)$$

https://doi.org/10.3390/electronics15081704

is generated with a controlled near-optimal geometry. Each vector is constructed as follows. A baseline value m is sampled as $m \sim \mathcal{N}(0,\, 0.5^2)$, and one index is assigned the minimum cost $\rho_{(1)} = m$. Since the entropic soft-min error $\mathcal{E}_\lambda(\rho)$ is invariant under additive shifts of the cost vector, the specific value of m does not affect the approximation error and is included only to avoid degenerate normalization. A set of k_{near} additional candidates is assigned near-optimal costs of the form

$$\rho_i = m + \delta_i, \qquad \delta_i \sim \text{Unif}[\delta_{\min}, \delta_{\max}],$$

while the remaining candidates are assigned larger excess costs sampled from

$$\delta_i \sim \text{Unif}[\delta_{\text{far,min}}, \delta_{\text{far,max}}].$$

Three geometry regimes are considered:

- Sharp geometry: one near-optimal competitor ($k_{\text{near}} = 1$), with small near gaps and large separation from the remaining candidates.
- Moderate geometry: several near-optimal candidates ($k_{\text{near}} = 4$) with intermediate separation.
- Flat geometry: many near-optimal candidates ($k_{\text{near}} = 10$) with small excess costs and reduced separation from the rest.

The specific parameter values used in the experiments are as follows:

- Sharp: $\delta_{\min} = 0.00$, $\delta_{\max} = 0.02$, $\delta_{\text{far,min}} = 1.00$, $\delta_{\text{far,max}} = 3.00$.
- Moderate: $\delta_{\min} = 0.00$, $\delta_{\max} = 0.06$, $\delta_{\text{far,min}} = 0.50$, $\delta_{\text{far,max}} = 2.00$.
- Flat: $\delta_{\min} = 0.00$, $\delta_{\max} = 0.12$, $\delta_{\text{far,min}} = 0.20$, $\delta_{\text{far,max}} = 1.00$.

This construction explicitly controls the near-optimal mass $N_\varepsilon(\rho)$ (introduced in Section 4) and the distribution of excess costs, allowing a direct assessment of how geometry affects the relaxation error.

For each generated cost vector, the relaxation error $\mathcal{E}_\lambda(\rho)$ is evaluated over a grid of inverse temperature parameters

$$\lambda \in [0.5, 120].$$

The calibrated inverse temperature λ^* used below is obtained by applying the conformal calibration rule of Section 5 to the full calibration sample in this experiment. This common value is then used in the Gibbs-weight and error-distribution panels.

Figure 1 illustrates the geometry-dependent behavior of the relaxation error. Figure 1a shows the decay of $\mathcal{E}_\lambda(\rho)$ as a function of λ, highlighting that the rate of convergence depends strongly on the geometry of the excess-cost vector. Figure 1b displays the distribution of sorted excess costs in the three regimes, making explicit the differences in near-optimal mass. Figure 1c shows the corresponding Gibbs weights at the calibrated inverse temperature, illustrating how geometry affects the concentration pattern. Finally, Figure 1d reports the distribution of the relaxation errors across regimes at the common calibrated value λ^*, showing that flatter geometries yield smaller errors.

These results show that the approximation error depends on the geometry of the excess-cost vector. In particular, larger near-optimal sets require smaller inverse temperatures to achieve a fixed tolerance, in agreement with the bounds in Section 4.

https://doi.org/10.3390/electronics15081704

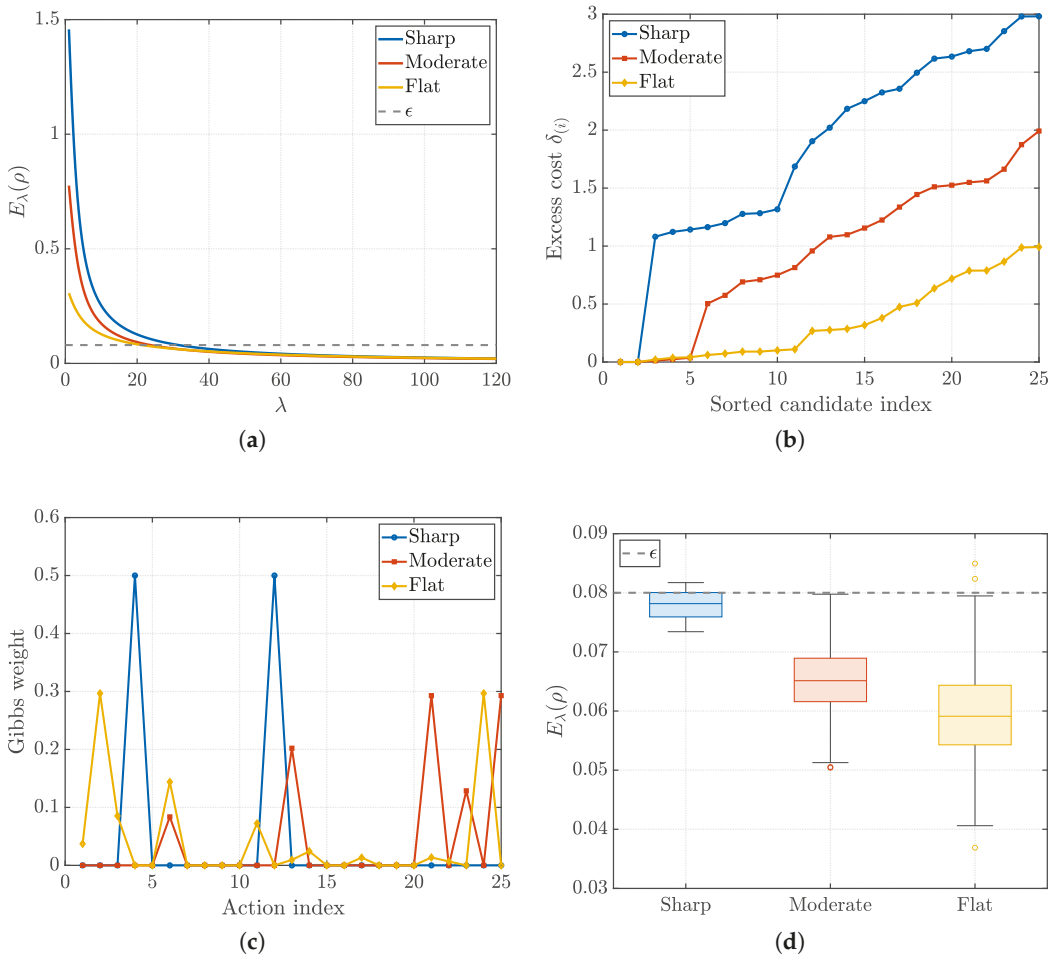


Figure 1. Experiment 1: geometry-dependent behavior of the entropic relaxation error across the three geometry regimes. (**a**) Relaxation error as a function of λ. Each curve corresponds to a representative cost vector from the sharp, moderate, and flat regimes, respectively. (**b**) Distribution of sorted excess costs $\delta_{(i)}$ in the three regimes. (**c**) Gibbs weights $p_i(\lambda;\rho)$ computed at the calibrated inverse temperature $\lambda = \lambda^*$, for representative cost vectors in each regime, plotted over the action index. (**d**) Boxplots represent the distribution of the relaxation error $\mathcal{E}_\lambda(\rho)$ over the test set for each regime, with $\lambda = \lambda^*$ fixed.

6.2. Experiment 2: Conformal Calibration of the Inverse Temperature

The second experiment evaluates the conformal calibration rule introduced in Section 5. The objective is to verify the empirical coverage guarantee and to compare the calibrated inverse temperature with several baseline selection rules in a contextual finite-action benchmark.

The data generation mechanism is contextual and geometry-controlled. Each instance consists of a context vector $x \in \mathbb{R}^d$ and an induced cost vector

$$\rho = (\rho_1, \ldots, \rho_N),$$

with $d = 6$ and $N = 20$. Calibration, test, and oracle samples are all drawn from the same nominal mixture of three geometry regimes with probabilities

$$(0.34, 0.33, 0.33).$$

The three regimes are as follows:

- Isolated: one exact minimizer, one near-optimal competitor, and the remaining actions separated by intermediate-to-large excess costs.
- Few-near: one exact minimizer, four near-optimal competitors, and the remaining actions distributed over intermediate and far excess-cost ranges.
- Many-near: one exact minimizer, ten near-optimal competitors, and the remaining actions concentrated closer to the minimum.

After assigning one exact minimizer, the near-optimal, intermediate, and far excess costs are sampled from regime-dependent uniform ranges. The parameter values used in the experiment are as follows:

- Isolated: near range $[0.35, 0.55]$, intermediate range $[0.90, 1.60]$, far range $[1.80, 3.20]$.
- Few-near: near range $[0.01, 0.04]$, intermediate range $[0.20, 0.55]$, far range $[0.65, 1.40]$.
- Many-near: near range $[0.00, 0.02]$, intermediate range $[0.03, 0.10]$, far range $[0.10, 0.35]$.

A small context-dependent perturbation is then added, after which the vector is recentered so that the minimum excess cost is zero and a common random offset is applied. Since the relaxation error is translation-invariant, this offset does not affect the operator-level calibration target.

Calibration is performed on a sample of size $n = 800$, test evaluation uses 6000 independent instances, and the oracle reference uses a much larger calibration sample of size 15,000. The target miscoverage level is $\alpha = 0.1$, the target tolerance is $\tau = 0.06$, and the inverse temperature is selected over an adaptively enlarged search grid.

The following inverse temperature selection rules are compared:

- ProposedConf: the conformal calibration rule defined in (43).
- GlobalFixed: a single fixed inverse temperature obtained as the average of the regime-specific conformal temperatures estimated on the calibration set:

$$\lambda_{\mathrm{GlobalFixed}} = \frac{1}{R} \sum_{r=1}^{R} \lambda_r^{\mathrm{conf}}, \qquad R = 3,$$

  where $\lambda_r^{\mathrm{conf}}$ denotes the conformal temperature obtained by applying the calibration rule to the subset of calibration instances belonging to regime r.
- MeanRisk: the smallest λ on the search grid such that the empirical mean relaxation error on the calibration set satisfies

$$\frac{1}{n} \sum_{j=1}^{n} \mathcal{E}_\lambda(\rho^{(j)}) \leq \tau.$$

- WorstCase: the deterministic choice

$$\lambda_{\mathrm{WorstCase}} = \frac{\log N}{\tau},$$

  obtained from the worst-case bound (35). This rule does not use calibration data and is therefore expected to be conservative.
- OracleCal: the same conformal calibration rule applied to a much larger calibration sample, used only as a reference to approximate the large-sample calibration target.

Table 1 reports the empirical coverage and approximation accuracy obtained by the different methods. Here, $q_{0.95}$ Error denotes the empirical 95th percentile of the test relaxation error.

https://doi.org/10.3390/electronics15081704

Table 1. Experiment 2: calibration results.

Method	Coverage	Mean Error	$q_{0.95}$ Error
ProposedConf	1.00000	0.043074	0.059861
GlobalFixed	0.45867	0.062963	0.095184
MeanRisk	0.53333	0.059268	0.088470
WorstCase	1.00000	0.043155	0.060000
OracleCal	1.00000	0.043074	0.059861

Table 1 shows that ProposedConf attains full empirical coverage in this benchmark and coincides with OracleCal up to numerical precision. The corresponding upper-tail error remains at the target tolerance level. By contrast, GlobalFixed and MeanRisk fail to control the upper tail of the relaxation error and exhibit substantial undercoverage. WorstCase also attains full coverage, but in this benchmark its performance is nearly indistinguishable from that of the conformal selector.

Figure 2 summarizes the behavior of the calibration procedure and the resulting performance. Figure 2a shows the empirical calibration curve $\widehat{q}_{1-\alpha}(\lambda)$, together with the tolerance level τ and the selected value λ^*. The monotonic decay of the curve and its intersection with τ illustrate the threshold structure underlying the conformal selection rule. Figure 2b reports the selected inverse temperatures for the different methods, showing that ProposedConf coincides with OracleCal and is numerically very close to the worst-case rule in this benchmark. Figure 2c displays the distribution of the test relaxation error for each method. The proposed calibration controls the upper tail of the error at the prescribed tolerance, whereas MeanRisk and GlobalFixed exhibit substantial violations. Figure 2d shows regime-wise coverage across methods. While ProposedConf maintains full coverage in all three regimes in this benchmark, GlobalFixed and MeanRisk exhibit marked heterogeneity across geometries. Figure 2e illustrates the relationship between the required inverse temperature and the top-two gap $\Delta_{(2)}(\rho)$, confirming that smaller gaps require larger values of λ, with additional variability induced by the near-optimal mass. Figure 2f shows the empirical distribution of the instance-wise required inverse temperatures. The selected value λ^* corresponds to the smallest value achieving the target level $1-\alpha$, as predicted by the conformal calibration rule.

In this benchmark, the conformal selector is numerically very close to the deterministic worst-case rule. This can be stated more precisely in terms of the population distribution of the instance-wise thresholds $\Lambda(\rho)$ introduced in Definition 5. The deterministic rule $\lambda_{\mathrm{WorstCase}} = \log N/\tau$ controls the tolerance for every cost vector and is attained, up to the asymptotic approximation of Section 4, by geometries with an isolated minimizer and a sufficiently separated near-optimal set, for which the relaxation error remains close to the worst-case scale $\log N/\lambda$. The conformal selector provides a meaningful improvement over this deterministic rule only when the upper $(1-\alpha)$-quantile of $\Lambda(\rho)$ lies strictly below $\log N/\tau$; that is, when such near-worst-case geometries occur with population probabilities smaller than α. In the nominal mixture of Experiment 2, difficult geometries still carry enough mass for the upper-tail calibration target to remain near the worst-case value. By contrast, the additional mixture-sensitivity study reported in Table 2 shows that the selector moves away from the worst-case scale when the population is sufficiently concentrated on the many-near regime.

https://doi.org/10.3390/electronics15081704

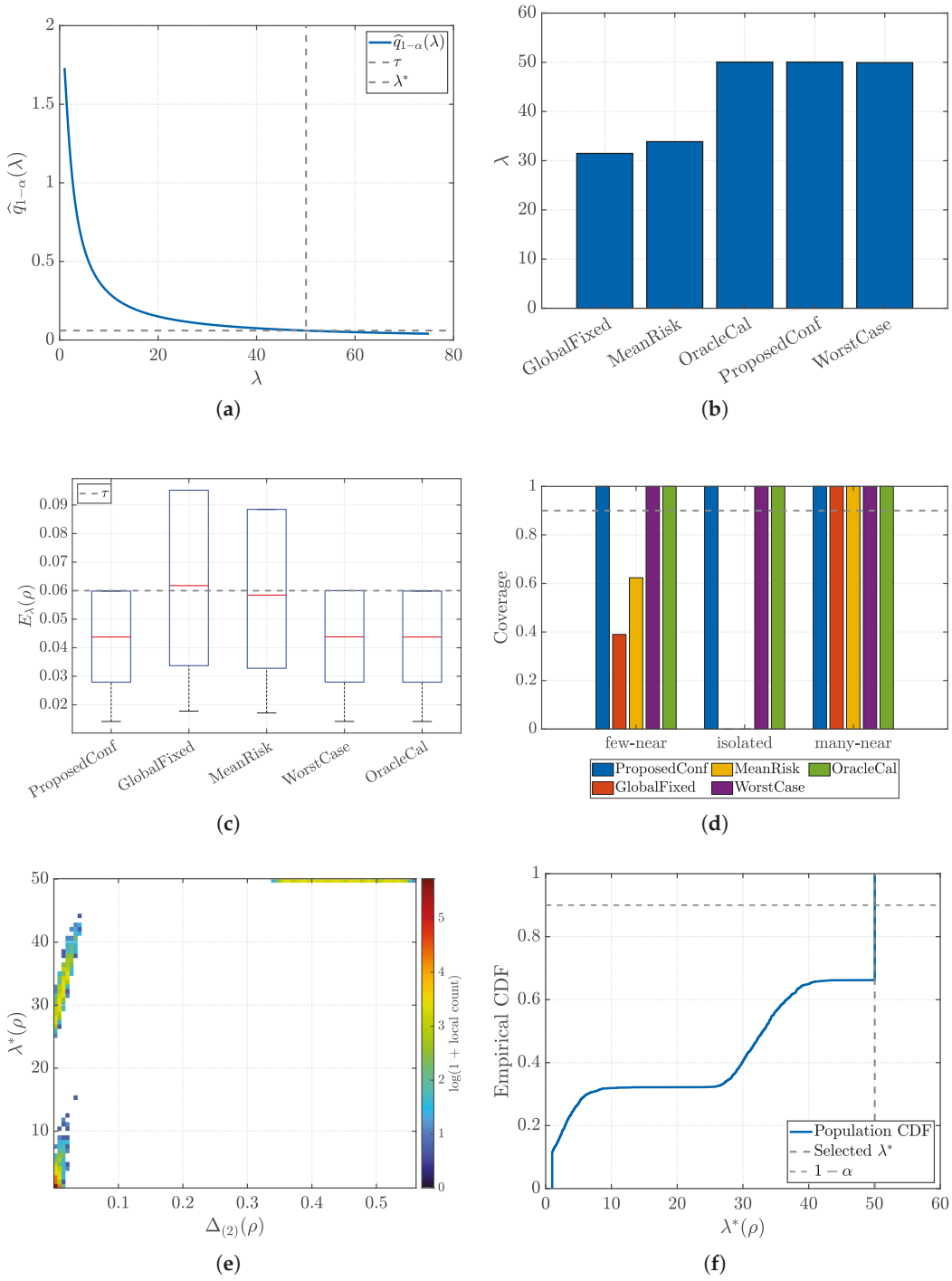


Figure 2. Experiment 2: empirical behavior of the conformal calibration rule in the contextual finite-action benchmark. (**a**) Calibration curve $\widehat{q}_{1-\alpha}(\lambda)$ computed on the pooled calibration sample. The horizontal line indicates the tolerance τ and the vertical line marks the selected value λ^*. (**b**) Selected inverse temperatures for the compared methods. (**c**) Boxplots of the test relaxation error $\mathcal{E}_\lambda(\rho)$ for the compared methods. The horizontal dashed line indicates the tolerance τ. (**d**) Regime-wise test coverage for all methods. Calibration is performed on the pooled nominal mixture, and evaluation is stratified according to the test regime. (**e**) Instance-wise relationship between geometry and required inverse temperature. Each point corresponds to a cost vector ρ, and $\lambda^*(\rho)$ denotes the smallest value of λ such that $\mathcal{E}_\lambda(\rho) \leq \tau$. (**f**) Empirical cumulative distribution function of the instance-wise required inverse temperatures $\lambda^*(\rho)$ in the population. The vertical line indicates the selected conformal value λ^*, and the horizontal line marks the target level $1 - \alpha$.

https://doi.org/10.3390/electronics15081704

To assess sensitivity to the nominal geometry mixture, we repeated the calibration experiment under two more extreme populations, namely $(0.85, 0.10, 0.05)$ and $(0.05, 0.10, 0.85)$, in addition to the baseline mixture $(0.34, 0.33, 0.33)$. In each case, calibration and test samples were drawn from the same mixture, the contextual generator and all numerical parameters were kept unchanged, and the experiment was repeated over 10 independent realizations. The resulting metrics are reported in Table 2. The observed pattern is asymmetric. When the population is shifted toward the isolated regime, the calibrated inverse temperature remains at the same value as in the baseline mixture, while the mean relaxation error increases. By contrast, when the population is strongly shifted toward the many-near regime, the calibrated inverse temperature decreases from 50.045 to 32.093 ± 1.144, and the empirical coverage moves close to the target level $1 - \alpha = 0.9$. Thus, in the present benchmark, the conformal selector remains at the near-worst-case scale as long as the upper-tail calibration target is dominated by difficult geometries, and decreases only when the geometry distribution is sufficiently concentrated on broad near-optimal sets. The reported $q_{0.95}$ values are included only as an auxiliary diagnostic, since the conformal calibration target is the upper $(1 - \alpha)$-quantile rather than the 0.95 quantile.

Table 2. Sensitivity of the conformal calibration rule to extreme nominal mixture weights in Experiment 2. Reported values are mean $\pm$ standard deviation over 10 independent realizations.

Mixture $(p_{\text{iso}}, p_{\text{few}}, p_{\text{many}})$	λ^*	Coverage	Mean Error	$q_{0.95}$ Error
$(0.34, 0.33, 0.33)$	50.045 ± 0.000	1.0000 ± 0.0000	0.04298 ± 0.00032	0.05986 ± 0.00000
$(0.85, 0.10, 0.05)$	50.045 ± 0.000	1.0000 ± 0.0000	0.05642 ± 0.00011	0.05986 ± 0.00000
$(0.05, 0.10, 0.85)$	32.093 ± 1.144	0.8979 ± 0.0124	0.03654 ± 0.00075	0.08462 ± 0.01296

The conformal guarantee relies on the exchangeability of calibration and test samples. To assess the effect of violating this assumption, we perform an additional experiment in which calibration and test data are drawn from different geometry regimes.

We consider shifted scenarios in which the calibration sample is drawn from a single regime, while the test sample is drawn from a different regime or from the nominal mixture. All other aspects of the setup are unchanged.

Table 3 reports the resulting coverage and approximation error. The results exhibit a clear and systematic pattern. When calibration is performed on geometries that are at least as challenging as those encountered at test time (for example, isolated $\rightarrow$ many-near), the method remains conservative and continues to satisfy the tolerance constraint. In contrast, when calibration is performed on more difficult geometries and evaluation is carried out on simpler ones (many-near $\rightarrow$ isolated), the selected inverse temperature becomes too small, leading to a substantial increase in the relaxation error and a collapse in coverage.

Table 3. Effect of distribution shift on conformal calibration. Calibration and test samples are drawn from different geometry regimes.

Scenario	Coverage	Mean Error	$q_{0.95}$ Error
nominal $\rightarrow$ nominal	1.00	0.043	0.060
isolated $\rightarrow$ many-near	1.00	0.025	0.033
many-near $\rightarrow$ isolated	0.00	0.520	0.526
isolated $\rightarrow$ few-near	1.00	0.043	0.050
many-near $\rightarrow$ nominal	0.30	0.271	0.526

These results are consistent with the theoretical analysis. In particular, the inverse temperature required to achieve a prescribed tolerance depends on the geometry of the

https://doi.org/10.3390/electronics15081704

excess-cost vector. Under distribution shift, the calibrated value may fail to control the upper tail of the relaxation error.

From a computational viewpoint, the proposed method separates an offline calibration stage from the online decision stage. In the straightforward implementation used here, calibrating over n cost vectors, an action dimension K, and a grid of m candidate inverse temperatures requires $O(mnK)$ operations to evaluate the relaxation errors and $O(mn \log n)$ operations to compute the empirical upper quantile by sorting at each grid point. Once λ^* has been selected, deployment uses a fixed inverse temperature and the online cost reduces to the standard Gibbs evaluation, namely $O(K)$ per decision, which is of the same order as for a fixed-temperature baseline. The geometric descriptors introduced in the analysis are not required at deployment time. Table 4 reports representative MATLAB wall-clock timings as the action dimension varies. Wall-clock times were measured in MATLAB on a Windows 64-bit machine equipped with an AMD Ryzen 9 8945HS processor (4.00 GHz) and 32 GB RAM. Reported values correspond to mean $\pm$ standard deviation over five independent runs, and online timings are normalized per decision using batches of 10^5 decisions. The results show that the calibration stage grows with the action dimension, whereas the online cost of ProposedConf remains essentially indistinguishable from that of the fixed-temperature baseline.

Table 4. Computational overhead of the conformal calibration module as a function of the action dimension K. Reported values are mean $\pm$ standard deviation over 5 independent runs.

K	Calibration Time (s)	Online Time/Decision (μs), ProposedConf	Online Time/Decision (μs), Fixed Baseline
10	0.440 ± 0.020	0.232 ± 0.013	0.213 ± 0.003
20	0.680 ± 0.009	0.302 ± 0.008	0.315 ± 0.004
50	0.962 ± 0.016	0.583 ± 0.014	0.616 ± 0.008
100	1.820 ± 0.021	0.968 ± 0.020	1.052 ± 0.028
200	2.875 ± 0.021	1.708 ± 0.034	1.866 ± 0.017

6.3. Experiment 3: Contextual Bandit with Sequential Risk

The third experiment studies the proposed calibration rule in a sequential decision setting. The goal is to assess how the calibrated inverse temperature affects both operator-level accuracy and downstream risk when action values are estimated.

The purpose of this experiment is illustrative rather than theorem-proving at the decision level. In particular, the conformal guarantee developed in Section 5 applies to the operator-level approximation error of the entropic relaxation and does not directly yield a bound on cumulative reward or long-term regret. Establishing such a link in multi-step decision problems would require additional assumptions on the environment dynamics, the propagation of value-estimation errors, and the sensitivity of the induced policy to approximation errors. Experiment 3 is formulated as a one-step contextual bandit with sequential risk, where downstream metrics describe how operator-level calibration is reflected in the induced decision behavior.

We consider a contextual finite-action bandit with one-step sequential risk. Each instance is defined by a context vector

$$x \in \mathbb{R}^d, \qquad d = 5,$$

and an associated estimated action-value vector

$$\widehat{Q}(x, \cdot) = \big(\widehat{Q}(x, 1), \ldots, \widehat{Q}(x, K)\big), \qquad K = 6.$$

https://doi.org/10.3390/electronics15081704

The corresponding cost vector is defined by

$$\rho_a(x) = -\widehat{Q}(x, a), \qquad a = 1, \ldots, K.$$

The policy induced by inverse temperature λ is the Gibbs policy

$$\pi_\lambda(a \mid x) = \frac{\exp(-\lambda \rho_a(x))}{\sum_{b=1}^{K} \exp(-\lambda \rho_b(x))}. \tag{47}$$

Larger values of λ yield a sharper policy, whereas smaller values yield a more diffuse one.

The six actions are organized into three semantic groups. Action 1 is a safe action, actions 2 and 3 are risky actions, and actions 4–6 are distractor actions with clearly inferior value in most contexts. The data-generating mechanism is a mixture of three geometry regimes with probabilities

$$(0.35,\ 0.40,\ 0.25):$$

- Safe-dominant: the safe action is clearly optimal and the risky actions are suboptimal but still competitive;
- Ambiguous-risk: a risky action appears slightly better than the safe action and the top gaps are small, so the local geometry is difficult and fragile;
- Flat: several actions are nearly tied, creating a broad near-optimal set.

For reproducibility, the distributions of the ground-truth action values $Q^\star(x, \cdot)$ are specified explicitly for each regime.

In the safe-dominant regime, the safe action has the highest expected value, while risky actions are suboptimal:

$$Q_1^\star \sim \mathcal{N}(1.10,\ 0.10^2), \quad Q_{2,3}^\star \sim \mathcal{N}(0.90,\ 0.06^2), \quad Q_{4:6}^\star \sim \mathcal{N}(0.40,\ 0.08^2).$$

In the *ambiguous-risk* regime, risky actions are competitive or slightly optimistic:

$$Q_1^\star \sim \mathcal{N}(1.00,\ 0.03^2), \quad Q_2^\star \sim \mathcal{N}(1.05,\ 0.03^2), \quad Q_3^\star \sim \mathcal{N}(1.01,\ 0.03^2),$$

while the remaining actions have lower values:

$$Q_{4:6}^\star \sim \mathcal{N}(0.55,\ 0.08^2).$$

In the *flat* regime, several actions are nearly tied:

$$Q_1^\star \sim \mathcal{N}(1.00,\ 0.03^2), \quad Q_2^\star \sim \mathcal{N}(0.99,\ 0.03^2), \quad Q_3^\star \sim \mathcal{N}(0.98,\ 0.03^2),$$

with remaining actions taking smaller values:

$$Q_4^\star \sim \mathcal{N}(0.95,\ 0.03^2), \quad Q_{5,6}^\star \sim \mathcal{N}(0.55,\ 0.06^2).$$

For each instance, a true action-value vector $Q^\star(x, \cdot)$ is sampled according to the regime, and a noisy estimate $\widehat{Q}(x, \cdot)$ is constructed by adding Gaussian noise with larger variance and positive bias on the risky actions, modeling optimistic overestimation.

The outcome of a selected action is realized rather than averaged deterministically. Safe and distractor actions return their nominal value, whereas risky actions produce either a positive outcome or an adverse event with regime-dependent probability. In particular, risky actions incur a higher failure probability when their nominal value exceeds one, and adverse realizations yield a substantial negative reward. Consequently, the inverse temperature directly affects risk exposure through policy concentration.

https://doi.org/10.3390/electronics15081704

Calibration, test, and oracle samples contain

$$n_{\text{cal}} = 800, \qquad n_{\text{test}} = 6000, \qquad n_{\text{oracle}} = 15000$$

instances, respectively. The target miscoverage level is $\alpha = 0.1$, the target tolerance is $\tau = 0.06$, and inverse temperatures are searched over the grid

$$\lambda \in [1, 120].$$

The compared inverse-temperature selection rules are the following:

- ProposedConf: the conformal selector defined in (43);
- FixedSoftmax: a fixed practical baseline with $\lambda = 12$;
- MeanRisk: the smallest grid value such that the empirical mean relaxation error on the calibration set is at most τ;
- WorstCase: the deterministic rule $\lambda = \log(K)/\tau$ derived from the worst-case bound;
- OracleCal: the conformal selector computed on a much larger calibration set;
- Greedy: the limiting sharply concentrated policy, approximated numerically by $\lambda = 10^6$.

The experiment is repeated over 25 independent random seeds, with independently generated calibration and test samples in each run. Reported results correspond to multi-seed averages together with their standard deviations.

Both operator-level and decision-level metrics are considered. Operator-level metrics include empirical coverage, mean relaxation error, and $q_{0.95}(E_\lambda)$. Decision-level metrics include mean realized return, adverse-event rate, average probability assigned to the true best action, and risky-action usage frequency.

Table 5 and Figure 3 show that ProposedConf maintains operator-level validity in the risky contextual setting. The upper-tail error remains at the target threshold, whereas MeanRisk and FixedSoftmax fail to control the upper tail. The selected conformal inverse temperature is stable across seeds and coincides with the oracle reference in this experiment:

$$\lambda_{\text{conf}} = 30.094, \qquad \lambda_{\text{oracle}} = 30.094.$$

This indicates that the conformal selector closely tracks the large-sample calibration target. Figure 3a shows a representative calibration curve $\widehat{q}_{1-\alpha}(\lambda)$ in the contextual setting. The monotonic decay and its intersection with the tolerance τ illustrate the threshold structure of the conformal selection rule. Figure 3b displays the instance-wise relationship between the required inverse temperature and the top-two gap $\Delta_{(2)}(\rho)$, confirming that calibration difficulty remains strongly geometry-dependent. Figure 3c,d report global operator-level metrics across seeds. ProposedConf maintains the desired coverage while controlling the upper-tail error at the prescribed level, whereas MeanRisk and FixedSoftmax exhibit clear violations. Figure 3e–g focus on the ambiguous-risk regime. They show that ProposedConf preserves operator-level validity while avoiding the overcommitment exhibited by the greedy policy. In contrast, FixedSoftmax achieves higher return only by violating the operator-level constraint.

Table 5. Experiment 3: multi-seed global results for the contextual bandit with sequential risk. Reported values are mean ± standard deviation over 25 random seeds.

Method	Coverage	$q_{0.95}(E_\lambda)$	Mean Return	Adverse Rate
ProposedConf	1.000 ± 0.000	0.05954 ± 0.00000	0.254 ± 0.018	0.304 ± 0.005
FixedSoftmax	0.0049 ± 0.0009	0.14771 ± 0.00009	0.284 ± 0.017	0.291 ± 0.005
MeanRisk	0.328 ± 0.011	0.06561 ± 0.00032	0.256 ± 0.020	0.303 ± 0.006
WorstCase	1.000 ± 0.000	0.06000 ± 0.00000	0.252 ± 0.021	0.304 ± 0.007

https://doi.org/10.3390/electronics15081704

Table 5. *Cont.*

Method	Coverage	$q_{0.95}(E_\lambda)$	Mean Return	Adverse Rate
OracleCal	1.000 ± 0.000	0.05954 ± 0.00000	0.259 ± 0.020	0.302 ± 0.006
Greedy	1.000 ± 0.000	0.00000 ± 0.00000	0.245 ± 0.019	0.308 ± 0.006

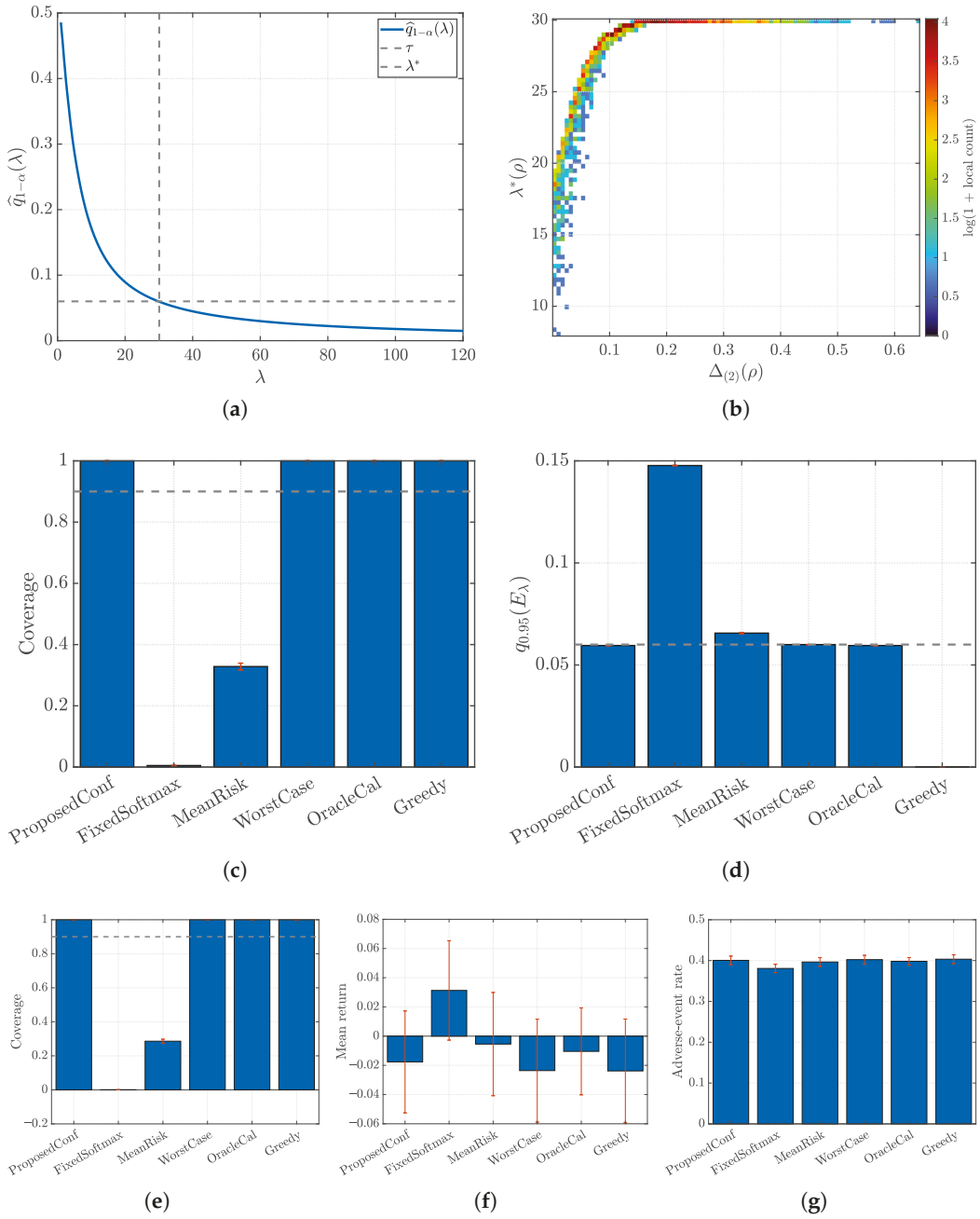


Figure 3. Experiment 3: contextual bandit with sequential risk. Calibration curve, geometry-dependent inverse temperature, and operator- and decision-level performance across regimes. (**a**) Representative conformal calibration curve $\widehat{q}_{1-\alpha}(\lambda)$ in the risky contextual setting. The horizontal line marks the tolerance τ, and the vertical line marks the selected λ^*. (**b**) Instance-wise geometry-to-calibration relationship: required inverse temperature $\lambda^*(\rho)$ versus top-two gap $\Delta_{(2)}(\rho)$. (**c**) Multi-seed global operator-level coverage. Error bars show one standard deviation across seeds. (**d**) Multi-seed global upper-tail operator error $q_{0.95}(E_\lambda)$. Error bars show one standard deviation across seeds. (**e**) Multi-seed metrics restricted to the ambiguous-risk regime: coverage. (**f**) Multi-seed metrics restricted to the ambiguous-risk regime: mean return. (**g**) Multi-seed metrics restricted to the ambiguous-risk regime: adverse-event rate.

Thus, the proposed rule is not a procedure designed to maximize return. Its role is to calibrate the sharpness of the induced Gibbs policy through operator-level control of the relaxation error. In the present experiment, this effect at the operator level translates into only moderate differences in adverse event rate, since the considered noise and failure regime does not produce large changes in exposure to risky actions across methods. Accordingly, the main conclusion is not that geometry-aware calibration uniformly improves downstream performance, but that it preserves the prescribed operator-level validity while interacting in a transparent way with the induced decision policy. Larger separations in realized return or adverse event frequency would be expected in regimes with stronger optimistic bias, higher estimation noise, or more pronounced asymmetry in the failure probabilities of the risky actions. A mathematical transfer from operator-level tolerance control to cumulative reward or long-term regret in multi-step decision problems is not established in the present analysis.

7. Concluding Remarks

This study investigated inverse-temperature calibration for entropic soft-min operators from a geometry-aware perspective. The analysis shows that the approximation error depends on the geometry of the excess-cost vector rather than only on support size.

Geometry-dependent bounds were derived to characterize how descriptors such as the top-two gap and the near-optimal mass affect the approximation accuracy. These results show that the inverse temperature required to achieve a given tolerance varies across populations with different near-optimal geometries.

A conformal calibration rule was introduced to select the inverse temperature with finite-sample distribution-free validity under exchangeability. The method selects the smallest value ensuring that a prescribed upper quantile of the relaxation error satisfies the target tolerance.

The numerical experiments are consistent with the theoretical analysis. The geometry-controlled benchmark shows the dependence of the relaxation error on the excess-cost geometry. The calibration experiments confirm that the proposed method attains the desired marginal coverage while avoiding the conservativeness of worst-case bounds and the failure of mean-based rules. The contextual bandit experiment shows that the same geometric dependence persists when the soft-min operator induces a decision policy.

The results indicate that inverse-temperature selection should be treated as a geometry-dependent calibration problem. The proposed method provides a principled and reproducible alternative to heuristic tuning rules.

The validity of the method depends on the geometry distribution represented in the calibration sample. Under distribution shift, the selected inverse temperature may fail to control the upper tail of the relaxation error. This reflects the role of exchangeability in the conformal guarantee.

The present framework is formulated at the population level, relies on the exchangeability assumption underlying the conformal guarantee, and develops its geometric analysis under the assumption of uniform weights. Extending the geometric descriptors and the associated bounds to weighted soft-min relaxations, where quantities such as the weighted near-optimal mass replace descriptors based on cardinality, remains an open direction, together with calibration depending on covariates, settings beyond exchangeability, and adaptive decision systems. In particular, dynamic multistep reinforcement learning problems introduce temporal dependence, distribution shift induced by the policy, and an additional gap between operator-level approximation error and long-term decision risk. Establishing a mathematical transfer from calibrated soft-min error control to cumulative reward or regret in such settings remains an open direction for future work.

https://doi.org/10.3390/electronics15081704

Likewise, embedding the present calibration scheme into standard multi-step reinforcement learning benchmarks such as Atari or MuJoCo would require fixing additional algorithmic choices concerning representation learning, policy optimization, exploration, and training dynamics. Such an evaluation is therefore left for future work.

Relatedly, integrating the present calibration rule with annealing schedules or learnable-temperature mechanisms within full reinforcement learning pipelines also remains a natural direction for future work.

More broadly, richer hierarchical decision architectures provide a natural application scenario for future extensions of the present framework. Recent application-oriented work on large language model-guided hierarchical games for intelligent battery coordination illustrates this broader class of settings [46].

Author Contributions: Conceptualization, J.E.S.; software, A.F.-F.; formal analysis, J.E.S.; investigation, J.E.S. and A.F.-F.; data curation, A.F.-F.; writing—original draft, J.E.S.; writing—review and editing, J.E.S. and A.F.-F.; supervision, J.E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the European Union's *DIGITAL Europe Programme* under Grant Agreement No. 101226207 (project *AI-SECRETT*), from the Spanish Government under Grant PID2024-156583OB-I00 (funded by MCIN/AEI/10.13039/501100011033), and from the Generalitat Valenciana under Grant CIGE/2024/195.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program.* **2005**, *103*, 127–152. [CrossRef]
2. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Number Pt. 1 in Berichte Über Verteilte Messysteme; Cambridge University Press: Cambridge, UK, 2004. Available online: https://books.google.es/books?id=mYm0bLd3fcoC (accessed on 16 March 2026).
3. Nishioka, A.; Kanno, Y. A feasible smoothing accelerated projected gradient method for nonsmooth convex optimization. *Oper. Res. Lett.* **2024**, *57*, 107181. [CrossRef]
4. Palomar, D.P. Convex Optimization Theory. In *Portfolio Optimization: Theory and Application*; Cambridge University Press: Cambridge, UK, 2025; pp. 491–538. Available online: https://portfoliooptimizationbook.com/slides/slides-convex-optimization-theory.pdf (accessed on 16 March 2026).
5. Blanchard, P.; Higham, D.J.; Higham, N.J. Accurately computing the log-sum-exp and softmax functions. *IMA J. Numer. Anal.* **2020**, *41*, 2311–2330. [CrossRef]
6. Calafiore, G.C.; Gaubert, S.; Possieri, C. Log-Sum-Exp Neural Networks and Posynomial Models for Convex and Log-Log-Convex Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 827–838. [CrossRef]
7. Zhang, S.; Tepedelenlioğlu, C.; Banavar, M.K.; Spanias, A. Max-consensus using the soft maximum. In Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–7 November 2013; pp. 433–437. [CrossRef]
8. Nowzari, A.; Rabbat, M.G. Improved Bounds for Max Consensus in Wireless Networks. *IEEE Trans. Signal Inf. Process. Netw.* **2019**, *5*, 305–319. [CrossRef]
9. Wainwright, M.J.; Jordan, M.I. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learn.* **2008**, *1*, 1–305. [CrossRef]
10. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2006. [CrossRef]
11. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2, Lake Tahoe, NV, USA, 5–10 December 2013; NIPS'13; pp. 2292–2300. [CrossRef]
12. Lefebvre, T.; Crevecoeur, G. On Entropy Regularized Path Integral Control for Trajectory Optimization. *Entropy* **2020**, *22*, 1120. [CrossRef]
13. Lefebvre, T.; Crevecoeur, G. Entropy Regularised Deterministic Optimal Control: From Path Integral Solution to Sample-Based Trajectory Optimisation. In *Proceedings of the 2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*; IEEE Press: Piscataway, NJ, USA, 2022; pp. 401–408. [CrossRef]

14. Williams, G.; Drews, P.; Goldfain, B.; Rehg, J.M.; Theodorou, E.A. Aggressive driving with model predictive path integral control. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1433–1440. [CrossRef]
15. Xu, H.; Xuan, J.; Zhang, G.; Lu, J. Trust region policy optimization via entropy regularization for Kullback–Leibler divergence constraint. *Neurocomputing* **2024**, *589*, 127716. [CrossRef]
16. Tao, F.; Wu, M.; Cao, Y. Generalized Maximum Entropy Reinforcement Learning via Reward Shaping. *IEEE Trans. Artif. Intell.* **2024**, *5*, 1563–1572. [CrossRef]
17. Bhole, A.; Filabadi, M.M.; Crevecoeur, G.; Lefebvre, T. Unifying Entropy Regularization in Optimal Control: From and Back to Classical Objectives via Iterated Soft Policies and Path Integral Solutions. *arXiv* **2025** , arXiv:2512.06109. [CrossRef]
18. Arriojas, A.; Adamczyk, J.; Tiomkin, S.; Kulkarni, R.V. Entropy Regularized Reinforcement Learning Using Large Deviation Theory. *Phys. Rev. Res.* **2023**, *5*, 023085. [CrossRef]
19. Labbi, S.; Mangold, P.; Tiapkin, D.; Moulines, E. On Global Convergence Rates for Federated Softmax Policy Gradient under Heterogeneous Environments. In Proceedings of the 29th International Conference on Artificial Intelligence and Statistics, Tangier, Morocco, 2–5 May 2026. Available online: https://openreview.net/forum?id=aPAq9U7MFm (accessed on 16 March 2026).
20. Vovk, V.; Gammerman, A.; Shafer, G. *Algorithmic Learning in a Random World*, 2nd ed.; Springer: Cham, Switzerland, 2022. [CrossRef]
21. Lei, J.; G'Sell, M.; Rinaldo, A.; Tibshirani, R.J.; Wasserman, L. Distribution-Free Predictive Inference for Regression. *J. Am. Stat. Assoc.* **2018**, *113*, 1094–1111. [CrossRef]
22. Balasubramanian, V.N.; Ho, S.S.; Vovk, V. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*; Morgan Kaufmann: San Francisco, CA, USA, 2014. [CrossRef]
23. Romano, Y.; Patterson, E.; Candes, E. Conformalized Quantile Regression. In *Proceedings of the Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32. Available online: https://proceedings.neurips.cc/paper_files/paper/2019/file/5103c3584b063c431bd1268e9b5e76fb-Paper.pdf (accessed on 16 March 2026).
24. Zhou, X.; Chen, B.; Gui, Y.; Cheng, L. Conformal Prediction: A Data Perspective. *arXiv* **2025**, arXiv:2410.06494. [CrossRef]
25. Cortes-Gomez, S.; Patiño, C.; Byun, Y.; Wu, S.; Horvitz, E.; Wilder, B. Utility-Directed Conformal Prediction: A Decision-Aware Framework for Actionable Uncertainty Quantification. *arXiv* **2025**, arXiv:2410.01767. [CrossRef]
26. Yeh, C.; Christianson, N.; Wierman, A.; Yue, Y. Conformal Risk Training: End-to-End Optimization of Conformal Risk Control. *arXiv* **2025**, arXiv:2510.08748. [CrossRef]
27. Huang, H.; Sharma, S.; Loquercio, A.; Angelopoulos, A.; Goldberg, K.; Malik, J. Conformal Policy Learning for Sensorimotor Control under Distribution Shifts. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 16285–16291. [CrossRef]
28. Flovik, V. Quantifying Distribution Shifts and Uncertainties for Enhanced Model Robustness in Machine Learning Applications. *arXiv* **2024**, arXiv:2405.01978. [CrossRef]
29. Lin, V.; Jang, K.J.; Dutta, S.; Caprio, M.; Sokolsky, O.; Lee, I. DC4L: Distribution shift recovery via data-driven control for deep learning models. In *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*; Abate, A., Cannon, M., Margellos, K., Papachristodoulou, A., Eds.; PMLR: Cambridge, MA, USA, 2024 ; Volume 242, pp. 1526–1538. Available online: https://proceedings.mlr.press/v242/lin24b.html (accessed on 16 March 2026).
30. Danesh, M.H.; Wabartha, M.; Pineau, J.; Lin, H.C. Mitigating Distribution Shifts: Uncertainty-Aware Offline-to-Online Reinforcement Learning. 2025. Available online: https://openreview.net/forum?id=0WqAnYWi7H (accessed on 1 March 2026).
31. Ellis, R.S. *Entropy, Large Deviations, and Statistical Mechanics*; Grundlehren der mathematischen Wissenschaften; Springer: New York, NY, USA, 1985. [CrossRef]
32. Amari, S.I. *Information Geometry and Its Applications*; Applied Mathematical Sciences; Springer: Tokyo, Japan, 2016. [CrossRef]
33. Donsker, M.D.; Varadhan, S.R.S. Asymptotic evaluation of certain markov process expectations for large time, I. *Commun. Pure Appl. Math.* **1975**, *28*, 1–47. [CrossRef]
34. Dembo, A.; Zeitouni, O. *Large Deviations Techniques and Applications*, 2nd ed.; Stochastic Modelling and Applied Probability; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
35. Dupuis, P.; Ellis, R.S. *A Weak Convergence Approach to the Theory of Large Deviations*; Wiley Series in Probability and Statistics; John Wiley & Sons: Hoboken, NJ, USA, 1997. [CrossRef]
36. Hartmann, C.; Richter, L.; Schütte, C.; Zhang, W. Variational Characterization of Free Energy: Theory and Algorithms. *Entropy* **2017**, *19*, 626. [CrossRef]
37. Wong, R. *Asymptotic Approximations of Integrals*; Academic Press: Cambridge, MA, USA, 1989. [CrossRef]
38. de Bruijn, N. *Asymptotic Methods in Analysis*; Bibliotheca Mathematica; Dover Publications: Garden City, NY, USA, 1981. Available online: https://books.google.es/books?id=_tnwmvHmVwMC (accessed on 16 March 2026).

39. Temme, N.M. Uniform asymptotic methods for integrals. *Indag. Math.* **2013**, *24*, 739–765. [CrossRef]
40. Oliveira, R.I.; Orenstein, P.; Ramos, T.; Romano, J.V. Split Conformal Prediction and Non-Exchangeable Data. *J. Mach. Learn. Res.* **2022**, *25*, 1–38.
41. Tibshirani, R.J.; Foygel Barber, R.; Candes, E.; Ramdas, A. Conformal Prediction Under Covariate Shift. In *Proceedings of the Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32. Available online: https://proceedings.neurips.cc/paper_files/paper/2019/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf (accessed on 16 March 2026).
42. Xu, C.; Xie, Y. Conformal prediction interval for dynamic time-series. In *Proceedings of the 38th International Conference on Machine Learning*; Proceedings of Machine Learning Research; Meila, M., Zhang, T., Eds.; PMLR: Cambridge, MA, USA, 2021; Volume 139, pp. 11559–11569. Available online: https://proceedings.mlr.press/v139/xu21h.html (accessed on 16 March 2026).
43. Zaffran, M.; Feron, O.; Goude, Y.; Josse, J.; Dieuleveut, A. Adaptive Conformal Predictions for Time Series. In *Proceedings of the 39th International Conference on Machine Learning*; Proceedings of Machine Learning Research; Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S., Eds.; PMLR: Cambridge, MA, USA, 2022 ; Volume 162, pp. 25834–25866. Available online: https://proceedings.mlr.press/v162/zaffran22a.html (accessed on 16 March 2026).
44. Vovk, V.; Nouretdinov, I.; Gammerman, A. Conformal e-testing. *Pattern Recognit.* **2025**, *168*, 111841. [CrossRef]
45. Angelopoulos, A.N.; Barber, R.; Bates, S. Online conformal prediction with decaying step sizes. In *Proceedings of the 41st International Conference on Machine Learning*; Proceedings of Machine Learning Research; Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F., Eds.; PMLR: Cambridge, MA, USA, 2024; Volume 235, pp. 1616–1630. Available online: https://proceedings.mlr.press/v235/angelopoulos24a.html (accessed on 16 March 2026).
46. Zou, Y.; Lin, Z.; Zhang, Q.; Liu, Z.; Xu, Z. Large language models enable semantic-guided hierarchical games for intelligent battery coordination. *Adv. Eng. Inform.* **2026**, *71*, 104312. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Information-Preserving Spiking for Accurate Time-Series Forecasting in Spiking Neural Networks

Jiwoo Lee [1] and Eun-Kyu Lee [2,*]

[1] Convergence Security Laboratory, Korea Automotive Technology Institute, Incheon 22851, Republic of Korea; jwlee3@katech.re.kr
[2] Department of Information and Telecommunication Engineering, Incheon National University, Incheon 22012, Republic of Korea
* Correspondence: eklee@inu.ac.kr

Abstract

Deep learning models have achieved high accuracy in forecasting problems, but at the cost of large computational energy demand. Brain-inspired spiking neural networks (SNNs) offer a promising, low-power alternative, yet their adoption for time-series forecasting has been limited by information loss from binary spikes and degraded performance in deeper networks. This paper proposes a fully spiking framework that bridges this gap by improving both the encoding and propagation of information in SNNs. The framework introduces a hybrid Delta-Rate encoding mechanism that captures both abrupt changes and gradual trends in time-series data, and a Mem-Spike mechanism that transmits analog membrane potential values to preserve fine-grained information between spiking layers. We further employ residual membrane connections to maintain signal flow in deep spiking networks. Using two public energy load datasets, our enhanced SNNs consistently outperform conventional spiking models, improving prediction accuracy by up to 61.6% and mitigating degradation in multi-layer networks. Notably, it narrows the gap to the selected deep learning baseline (LSTM), achieving comparable accuracy in some settings while requiring only about 10% of the estimated inference energy of that baseline under a common operation-level model. These results show that, within the empirical scope considered here, enhanced conventional SNNs can improve time-series forecasting accuracy while retaining favorable estimated efficiency.

Keywords: spiking neural network; time-series forecasting; load forecasting; prediction; accuracy; encoding; information loss

1. Introduction

Deep learning has achieved remarkable progress across a wide range of domains. In computer vision, large-scale data-driven convolutional models and residual learning have driven rapid performance gains [1,2], while in natural language processing, self-attention-based Transformers and pretrained language models have become the de facto standard [3,4]. Deep learning has also been actively adopted in scientific and industrial applications such as high-resolution weather forecasting and power load forecasting [5,6]. As new architectures emerged, model depth and architectural complexity have steadily increased [2,3], driving advances in large-scale parallel and distributed learning [7] and highlighting the importance of energy-efficient model design and compression under constrained computation and power budgets [8–10]. In time-series load forecasting, achieving

high predictive accuracy remains essential [6,11,12], and efficient operation under resource constraints has become an increasingly important requirement.

Against this backdrop, neuromorphic computing has attracted considerable attention as a promising approach for improving efficiency beyond conventional deep learning [13,14]. Neuromorphic platforms such as IBM TrueNorth, Intel Loihi, and the SpiNNaker system have been introduced, all of which are designed around spiking neural networks (SNNs) [15–17]. SNNs draw inspiration from event-driven processing in biological neural circuits and process information using discrete spikes rather than continuous-valued activations [18–21]. This spike-based communication is often regarded as a closer abstraction of neural activity and can offer advantages in terms of low power consumption and reduced computational demand [13,15,16]. Accordingly, SNNs and neuromorphic hardware are attractive candidates for domains where energy efficiency and real-time processing are critical, including robotics and edge intelligence [13,22].

However, existing works on SNNs have focused predominantly on recognition and classification tasks, such as image classification, speech recognition, and event-based perception [20–22]. For instance, SNNs have been extensively studied for image classification via training and ANN-to-SNN conversion pipelines [23–25], and low-power speech keyword spotting has also been explored with deep spiking models [26]. In contrast, applications of SNNs to forecasting (regression) tasks based on time-series data remain relatively limited [20–22]. Meanwhile, time-series forecasting in real-world and industrial settings is still dominated by deep learning approaches, including recurrent, convolutional, and transformer-based architectures [11,12]. The central problem addressed in this paper is that, although conventional SNNs offer advantages in terms of energy efficiency, they remain less competitive than standard deep learning models in terms of forecasting accuracy.

This inferior predictive performance mainly stems from two structural factors. First, conventional spike-based encoding mechanisms have inherent limitations. While typical deep learning models directly consume continuous time series as input, SNNs must convert signals into discrete spike trains, during which fine-grained variations may not be sufficiently represented as spikes [18,19,27]. Since forecasting tasks are highly sensitive to small changes, such information loss at the input stage can cause non-trivial degradation in predictive accuracy [27]. Second, SNN neurons transmit signals only when the membrane potential surpasses a threshold, which can lead to increasingly sparse spike activity as the network becomes deeper. This spike sparsity may induce information loss, destabilize signal propagation during training, and degrade overall performance, especially in deep SNN architectures where spikes can become excessively sparse in intermediate layers.

The goal of this study is to mitigate these issues and, while preserving a conventional LIF-based SNN architecture, to raise accuracy for time-series prediction to a practically useful level. To this end, this study designs a information-preserving spiking framework that includes three core mechanisms. The term "information-preserving" is used in an empirical and operational sense, meaning that the proposed framework better retains prediction-relevant temporal signals than conventional spike-only encoding and propagation, rather than implying a formal information-theoretic guarantee.

First, we propose a *Delta-Rate Encoding* that combines a delta-based encoding, which is sensitive to abrupt changes, with a rate-based encoding, which repeatedly captures smooth trends, in order to effectively capture diverse patterns in time-series data. By combination, the new spike encoding mechanism can represent both rapidly changing regions and stable segments in the time series data and convert them into spike sequences to better preserve predictive patterns, thereby alleviating the limitations of existing encoding mechanisms.

Second, to mitigate information loss caused by spike sparsity, we propose a *Mem-Spike* mechanism that propagates real-valued signals instead of binary spikes ("0" and "1") [28].

https://doi.org/10.3390/electronics15081597

In the mechanism, the neuronal membrane potential is directly forwarded to the next layer, enabling more continuous information representations and more stable predictions. This mechanism is compatible with existing LIF and RLIF layers and extends conventional SNNs so that they can transmit richer time-series information than when using purely binary spikes, thereby improving forecasting accuracy.

Last, to address information loss in deep network architectures more effectively, we propose a *Residual-Membrane Connection*. It compensates for missing information by propagating neuronal membrane potentials through skip connections, instead of forwarding only the layer outputs to subsequent layers. This approach helps maintain the flow of neural information even in deep networks and contributes to more reliable signal propagation during training. Importantly, Residual-Membrane is not a new self-attention-based architecture, but rather a simple add-on module that can be added on top of existing LIF- and RLIF-based SNN layers to improve forecasting performance in deeper conventional SNNs while preserving their original architectural form.

This study validates the effectiveness of the proposed mechanisms through extensive load forecasting experiments using energy and electricity datasets. Experimental results demonstrate that the enhanced SNN models reduce the Relative Squared Error (RSE) by up to 61.6% compared to conventional LIF and RLIF models, while achieving performance comparable to LSTM in long-term prediction horizons. Additionally, energy consumption estimates reflecting neuromorphic hardware characteristics confirm approximately 91% energy savings over traditional ANNs, enabling high-precision time-series analysis. This framework presents a practical alternative for real-time time-series forecasting in resource-constrained settings.

We note that this work focuses on improving input encoding and membrane-potential propagation for layered and recurrent SNNs based on LIF and RLIF neurons that do not employ self-attention [19,29], rather than on the family of spiking transformers that incorporate computationally expensive modules such as self-attention [3,30]. The relationship and differences between our approach and self-attention-based spiking models are discussed in detail in Section 2.

2. Related Works

2.1. Time-Series Forecasting

Time-series forecasting is an analytical approach that estimates future values from past observations and plays a central role in a wide range of domains, including finance, weather, load forecasting, and traffic demand [31]. In particular, load forecasting is directly related to power system operation, capacity planning, and the design of demand response strategies, and therefore requires high predictive accuracy and stability [32]. Early research on time-series forecasting was mainly based on statistical methods such as autoregressive integrated moving average (ARIMA) models and Gaussian processes [33,34]. These classical models offer relatively high interpretability and are effective for modeling short-range linear patterns, but they show limitations when dealing with complex time series that exhibit strong nonlinearity and long-term dependencies.

With the development of deep learning, time-series forecasting has increasingly adopted neural architectures, including recurrent models such as RNNs [35], LSTMs [36], and GRUs [37], as well as temporal convolution models (e.g., TCN) [38]. In addition, domain-specific forecasting architectures such as LSTNet have also been explored [11]. CNN-based models are particularly strong at extracting local temporal patterns through one-dimensional convolutions [38], while RNNs, LSTMs, and GRUs are widely used to model both short- and long-term dependencies by processing sequential inputs step by step [35–37]. More recently, a three-channel LSTM–CNN hybrid model has been reported

https://doi.org/10.3390/electronics15081597

to improve short-term power load forecasting performance [39]. Furthermore, transformer-based models such as Informer, Autoformer, and iTransformer have been developed to capture long-term dependencies over extended time horizons while alleviating computational complexity through techniques such as sparse attention and series decomposition [12,40,41]. Although these ANN-based models can achieve very high forecasting performance when sufficient computational resources are available, the rapid growth in the number of parameters and in the required multiply–accumulate operations makes it difficult to deploy them as they are on edge devices or in energy-constrained environments [8–10].

As ANN-based time-series forecasting methods have matured, there has recently been growing interest in applying SNNs to time-series forecasting [42–44]. Lv et al. propose a unified framework that selects forecasting backbones and converts them into spiking counterparts (Spike-TCN, Spike-RNN, iSpikformer), and report competitive forecasting accuracy using dedicated encoders and temporal alignment strategies [42]. TS-LIF introduces a dual-compartment neuron designed for time-series forecasting and reports improved forecasting accuracy over conventional LIF-based SNNs when combined with CNN, RNN, and transformer [43]. In addition, works such as SpikeSTAG integrate GNNs and SNNs into a cooperative architecture and introduce composite modules that combine spiking self-attention with LSTM-based gating, and report strong SNN-based forecasting performance on multivariate time-series forecasting problems with complex spatio-temporal dependencies [44].

These studies share the common characteristics: (1) they replace already sophisticated ANN forecasting architectures such as TCNs, RNNs, transformers, and GNNs with spiking activations, or (2) they define new neuron models, such as TS-LIF, that extend conventional LIF/RLIF neurons and thereby alter the network mechanism itself. In this sense, they follow a different design philosophy from "conventional SNNs" composed purely of standard LIF/RLIF neurons. In contrast, this paper focuses on improving encoding and membrane-potential propagation mechanisms in SNNs composed solely of conventional LIF/RLIF neurons—without introducing complex modules such as self-attention, gated recurrent units, or dual-compartment neurons—to enhance forecasting accuracy. Therefore, ANN-backbone-type and extended-neuron-type SNN models such as those described above are conceptually related but are not considered within the scope of this work.

2.2. SNN Training

SNNs are artificial neural networks that mimic biological neural systems and can reduce effective computation under sparse spiking and event-driven execution, particularly on neuromorphic hardware [13,18,22]. Unlike DNNs, which operate on continuous-valued activations, SNNs transmit information via discrete spike signals and perform computation only when events occur, exhibiting an event-driven processing paradigm [13,18]. However, because the spike generation function of a neuron is discrete and non-differentiable, standard backpropagation used in conventional neural networks cannot be directly applied, which poses a fundamental challenge for training SNNs.

To address this issue, two representative approaches that combine ANNs and SNNs have been proposed. The first is ANN-to-SNN conversion [23,45]. In this approach, a high-performing ANN model is trained first, and then its weights are mapped onto a spiking neuron architecture to obtain an SNN, with the goal of maintaining ANN-level performance while benefiting from the energy efficiency of spike-based computation [23]. Recent studies have proposed various techniques to reduce conversion errors, including threshold adjustment tailored to activation distributions [45], quantization-aware training [46], and layer-wise calibration [47]. The second approach is direct training using surrogate gradients [24,25,48]. This method replaces the non-differentiable spike generation func-

https://doi.org/10.3390/electronics15081597

tion with a differentiable surrogate function during training, enabling backpropagation, while preserving discrete spike-based operations in the forward pass. In this way, gradient propagation becomes feasible even in deep SNNs, and a number of works have reported competitive performance on benchmarks such as image classification.

ANN-to-SNN conversion and surrogate-gradient training have broadened the range of trainable SNNs, and the LIF/RLIF-based SNNs used in this paper are trained via backpropagation with surrogate gradients. Nonetheless, most prior work has focused on tasks that require discrete outputs, such as image classification or event-based detection, while comparatively less attention has been paid to the structural design needed to predict continuous time series with high accuracy, where both inputs and outputs are continuous-valued regression signals [20,21].

This paper builds on these training techniques and focuses on improving forecasting accuracy in conventional LIF/RLIF-based SNNs by designing time-series-oriented encoding (Delta-Rate) and membrane-potential propagation (Mem-Spike, Residual-Membrane) mechanisms. In other words, our goal is to demonstrate that conventional SNNs composed solely of standard LIF/RLIF neurons can achieve competitive performance on time-series forecasting tasks through structural enhancements that are complementary to existing SNN training methods, without introducing complex modules such as self-attention, gated recurrent units, or dual-compartment neurons.

3. Problem Analysis

This section summarizes the key bottlenecks that limit the performance of SNNs on forecasting tasks and provides background analysis to derive design requirements for the proposed methods. In particular, we focus on signal information loss caused by the binary firing mechanism of SNNs and on spike sparsity arising from subthreshold membrane potentials [27,49]. These structural limitations of SNNs make it difficult to capture fine-grained variations in regression forecasting and cause a rapid decrease in effective signal propagation as the network becomes deeper [49–51].

3.1. Problem Statement 1: Information Loss

Although conventional SNNs more closely mimic the operation of the biological brain than traditional deep learning models, they still exhibit several limitations [18,22]. Most notably, their use of purely binary spike signals makes it impossible to process stimulus intensity in a differentiated manner. All stimuli are represented by the same spike value ("1"), which leads to information loss and degrades performance on tasks that involve continuous-valued data, such as regression forecasting.

In typical SNNs, the forward propagation of the membrane potential can be written as

$$U[t+1] = U[t] + \mathrm{Syn}[t]S[t],$$

where $U[\cdot]$ denotes the membrane potential, $S[\cdot]$ the spike value, and $\mathrm{Syn}[\cdot]$ the synaptic weights. The spike $S[\cdot]$ is defined as 1 if $U[\cdot]$ exceeds the threshold U_{thr}, and 0 otherwise. In this formulation, the input $S[\cdot]$ is strictly binary, so the information transmitted to the next neuron cannot fully reflect the variations present in the current input.

By contrast, the forward propagation in a conventional DNN can be expressed as

$$O[t] = W[t]X[t],$$

where the output $O[t]$ varies continuously with the magnitude of the input $X[t]$. In SNNs, however, this dependency is lost due to the binary nature of spikes, resulting in information loss. In regression forecasting tasks, where the model must respond sensitively to small

https://doi.org/10.3390/electronics15081597

changes in the input, such binary signaling may degrade forecasting accuracy. Therefore, a new mechanism is required that can more finely encode stimulus intensity.

More concretely, conventional SNN models process all inputs using the same spike value regardless of stimulus magnitude. In a loss function such as the mean squared error (MSE), gradients are computed solely based on the difference between the predicted value y_{pred} and the ground-truth value y_{true}. When the stimulus $X[t]$ is zero, updates depend on the previous state, whereas when $X[t] = 1$, all neurons emit the same spike value (1) and the weights are updated accordingly. As a result, the network exhibits the same response irrespective of stimulus strength and fails to distinguish between strong and weak stimuli.

This leads to the issue that subtle changes in the stimulus may be ignored by the SNN. In biological neural systems, a stronger stimulus typically induces more spikes, and the number of spikes conveys information proportional to the stimulus strength [19,27]. In contrast, conventional SNNs struggle to realize such differentiated processing, which causes information loss and ultimately degrades the predictive performance of the network.

3.2. Problem Statement 2: Low Spike Frequency

Another fundamental problem of SNNs is their low spike frequency [21,48,49]. In SNNs, a spike is generated only when a neuron's membrane potential exceeds a threshold. However, strong stimuli that repeatedly drive the membrane potential above the threshold are relatively rare; in time-series data in particular, weak stimuli often accumulate slowly until they eventually produce a spike. Consequently, during training, neurons tend to fire infrequently, and the network fails to maintain sufficiently active neural dynamics to effectively transmit information. A further important issue is the decrease in spike frequency as the model becomes deeper [49–51]. As signal propagation continues through successive layers, spike events become increasingly rare, and signals fail to reach deeper layers. Thus, as network depth increases, the number of active neurons decreases sharply. This phenomenon severely limits the ability to learn complex patterns, especially in problems where depth is critical for modeling capacity.

One naive way to increase the spike frequency is to lower the firing threshold [21]. However, reducing the threshold impairs the network's ability to reflect stimulus strength. With a lower threshold, neurons fire even in response to small stimuli, making it difficult to distinguish between different stimulus intensities and potentially degrading model performance [19,27]. Therefore, simply lowering the threshold does not provide a fundamental solution. In summary, the practical difficulty in deep forecasting networks is not determined only by the number of spikes. Even when firing rates are not extremely low, prediction quality may still degrade if the emitted spikes carry insufficient information about fine-grained input variations or if useful signals become unstable across depth. Accordingly, the key design goal is not merely to increase firing activity, but to improve the informativeness of spike-mediated communication and maintain stable signal propagation in deep architectures.

4. Methodology

4.1. Information-Preserving Spiking Framework

This section proposes a new framework that can improve the accuracy of time-series forecasting in SNNs by capturing both abrupt changes and subtle variations in time-series data. Technically, the framework is designed to (i) accurately encode time-series inputs into spike signals, (ii) reduce information loss, and (iii) mitigate depth-related degradation in signal propagation even in deep network architectures. The overall process consists of the following three core mechanisms, also shown in Figure 1:

1. *Delta-Rate Encoding*: To convert input time-series data into spike trains, we adopt a hybrid encoding mechanism that can capture both abrupt and gradual changes. Abrupt changes in the input signal are sensitively detected through delta encoding, while gradual variations below a threshold are accumulated and captured through rate encoding.
2. *Mem-Spike Mechanism*: To address the loss of fine-grained value information during spike conversion, we introduce a Mem-Spike mechanism that forwards the membrane potential of a neuron to the next layer only when that neuron emits an output spike. In this way, the membrane potential complements information that cannot be fully represented by spike events alone, enabling better preservation of information for tasks such as time-series forecasting that require continuous-valued outputs.
3. *Residual-Membrane Connection*: To reduce information loss in deep SNN architectures, we employ a Residual-Membrane mechanism that propagates the membrane potential at the input of a layer directly to its output via a skip connection. Unlike standard residual learning, which connects layer outputs, this mechanism preserves the membrane potential state at the input and reflects it in subsequent neurons alongside the layer output, thereby stabilizing the retention of early input information.

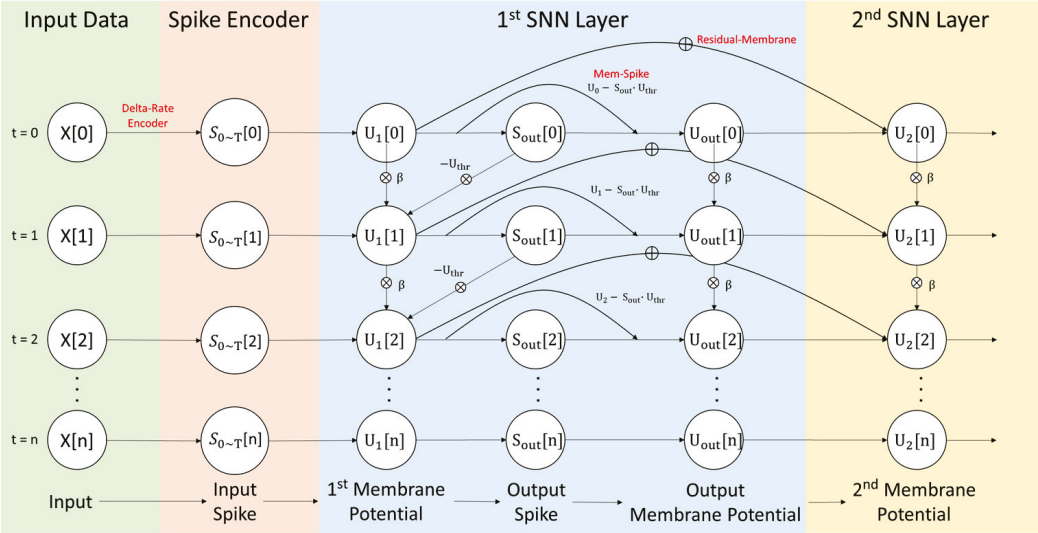


Figure 1. The information-preserving spiking framework includes three core mechanisms.

The overall data flow is as follows. The input time-series data are first transformed into spike sequences through Delta-Rate encoding. As the resulting spikes propagate through each layer, the Mem-Spike mechanism is applied to minimize information loss. When stacking the network more deeply, the Residual-Membrane mechanism complements signal transmission by forwarding the input membrane potential to subsequent neurons, thereby maintaining the flow of neural information. Finally, at the output layer, continuous target values are reconstructed to perform time-series forecasting.

4.2. Delta-Rate Encoding

Time-series data typically exhibit a mixture of abrupt changes and gradual trends. Conventional single encoding mechanisms have limited ability to capture such diverse patterns simultaneously. To address this issue, this study proposes a Delta-Rate encoding mechanism that combines delta encoding and rate encoding. The proposed method is designed to capture the full range of variability in time-series data by responding sensitively to sharp changes while gradually accumulating slower variations.

https://doi.org/10.3390/electronics15081597

Delta spike transformation. We first compute the difference between adjacent elements along the feature dimension for a given input x. Whenever the difference exceeds a predefined threshold θ in either direction, a spike is generated. This process can be formulated as:

$$\Delta x^{(i)}(t) = x^{(i)}(t) - x^{(i-1)}(t),$$

$$S^{(i)}_{\text{delta}}(t) = \begin{cases} +1, & \Delta x^{(i)}(t) \geq \theta \\ -1, & \Delta x^{(i)}(t) \leq -\theta \\ 0, & \text{otherwise} \end{cases}$$

where $x^{(i)}(t)$ denotes the input value of feature i at time t, $\Delta x^{(i)}(t)$ is the change between adjacent features, and θ is the threshold that determines spike generation. For the first feature, no previous element exists for comparison, so the spike is set to zero.

Rate accumulation update. After delta transformation, we apply a rate-based accumulation update to reflect gradual variations that do not cross the threshold. The rate update is defined in terms of a current state $u(t)$ as:

$$u(t+1) = \begin{cases} u(t) - \theta, & \text{if } u(t) \geq \theta \\ u(t) + x(0), & \text{otherwise} \end{cases}$$

where $x(0)$ is the initial input value. In particular, if a spike has been generated in the previous step due to crossing the threshold, the next step forcibly accumulates the input as $u(t) + x(0)$. This allows small changes in the input to be gradually integrated over time and prevents information from being lost due to single, isolated spike events.

Delta-Rate hybrid encoding. The proposed encoding mechanism combines the advantages of delta encoding, which reacts immediately to abrupt changes, with those of rate encoding, which accumulates and reflects continuous variations, so that diverse patterns in time-series data can be represented with reduced information loss.

More concretely, delta encoding operates on adjacent differences in the input sequence,

$$d^{(t)} = c^{(t)} - c^{(t-1)},$$

and binarizes them using a threshold θ. When $d^{(t)} \geq \theta$, it records an immediate $+1$ spike, and when `off_spike` = `true`, it also records a -1 spike when $d^{(t)} \leq -\theta$, thereby capturing sharp upward and downward transitions.

In contrast, the core idea of rate encoding is to accumulate subthreshold changes at each step according to

$$c^{(t+1)} \leftarrow c^{(t)} + X,$$

so that once sufficient evidence has been accumulated and $c^{(t)} \geq \theta$, a *delayed spike* naturally occurs. This mechanism allows subtle variations to be reflected gradually over time. A normalization step such as:

$$c^{(t+1)} \leftarrow c^{(t)} - \theta$$

is used as an auxiliary technique to mitigate chains of repeated spikes.

As summarized in Algorithm 1, starting from $c^{(1)} \leftarrow X$, each step $t = 1, \ldots, S$ proceeds as follows: (i) compute $d^{(t)}$ and generate $\Delta^{(t)} \in \{-1, 0, +1\}$, which is stored as the delta spike at step t; (ii) if no spike is generated, update $c^{(t+1)} \leftarrow c^{(t)} + X$ to accumulate subthreshold inputs, whereas if the threshold is exceeded, apply appropriate normalization and move on to the next step. The final output is the time-stacked spike sequence $\Delta S = \{\Delta^{(t)}\}_{t=1}^{S}$. In this way, the sharp-change sensitivity of delta encoding and the persis-

https://doi.org/10.3390/electronics15081597

tence of rate encoding work in a complementary manner, jointly mitigating information loss due to spike encoding and signal vanishing caused by spike sparsity in continuous-valued forecasting tasks.

Algorithm 1 Delta–Rate encoding

Require: input tensor X (last dim $= F$), threshold $\theta > 0$, steps $S \in \mathbb{N}$, option `off_spike` $\in \{\texttt{true}, \texttt{false}\}$
Ensure: `delta_spike_outputs`

$c \leftarrow X$
$flag \leftarrow \mathbf{0}_{\text{bool}}$
$\Delta S \leftarrow [\,]$
for $t = 1$ to S **do**
  $\Delta \leftarrow \mathbf{0}$
  $d \leftarrow c[\ldots, 2{:}F] - c[\ldots, 1{:}F{-}1]$
  $\Delta[\ldots, 2{:}F] \leftarrow 1[d \geq \theta]$
  if `off_spike` $=$ `true` **then**
    $\Delta[\ldots, 2{:}F] \leftarrow \Delta[\ldots, 2{:}F] - 1[d \leq -\theta]$
  $\Delta[\ldots, 1] \leftarrow 0$
  append Δ to ΔS
  if $t = S$ **then**
    break
  $next \leftarrow \text{where}\big(flag,\ c + X,\ \text{where}(c \geq \theta,\ c - \theta,\ c + X)\big)$
  $flag \leftarrow \text{where}\big(flag,\ \mathbf{0},\ c \geq \theta\big)$
  $c \leftarrow next$
return stack(ΔS)

4.3. Mem-Spike Mechanism

In SNNs, a neuron's membrane potential evolves in response to input stimuli, and a spike is emitted when the membrane potential exceeds a threshold (v_{th}). This study is based on the Leaky Integrate-and-Fire (LIF) model, where the membrane potential is updated in a time-discretized form as:

$$U[t] = \beta U[t-1] + I[t] - S[t-1]v_{th},$$

where $U[t]$ is the membrane potential at time t, β is the leak coefficient, $I[t]$ is the input current, and $S[t-1]$ indicates whether a spike occurred at the previous step. The term $-S[t-1]v_{th}$ accounts for the reset effect from the previous time step that is carried into the current update. The spike event is defined as:

$$S[t] = \begin{cases} 1, & \text{if } U[t] \geq v_{th} \\ 0, & \text{otherwise.} \end{cases}$$

When a spike occurs, we adopt a soft reset instead of a hard reset, subtracting the threshold from the membrane potential after the spike decision:

$$U[t] \leftarrow U[t] - S[t]v_{th}.$$

Soft reset preserves residual membrane information after firing, enabling the neuron to respond more sensitively to continuous input patterns over time.

In conventional SNNs, only the binary spike indicator $S[t]$ is forwarded to the next layer, which limits the ability to reflect fine-grained variations in the input signal or detailed membrane-potential states. To address this limitation, this study introduces the Mem-Spike

https://doi.org/10.3390/electronics15081597

mechanism, which emits membrane information *gated by spike events*, i.e., it forwards the (soft-reset) membrane potential only at firing times. The output signal in it is defined as:

$$O[t] = S[t] \times U[t].$$

With this formulation, the output becomes 0 when no spike occurs ($S[t] = 0$), whereas when a spike occurs ($S[t] = 1$), the post-reset membrane potential is directly emitted as the output. Note that $U[t]$ in $O[t]$ denotes the *post-reset* membrane potential.

The Mem-Spike mechanism is particularly beneficial for forecasting problems, such as those involving time-series data, where subtle input changes are important. Compared with conventional binary spike propagation, Mem-Spike increases sensitivity to input variations, reduces information loss, and can improve forecasting performance. The Mem-Spike output generation procedure is summarized in Algorithm 2. Because Mem-Spike forwards a real-valued membrane payload, it should be interpreted as a pragmatic model-level mechanism rather than a universally hardware-native spike primitive; more hardware-constrained deployments may require quantized or approximate variants of this transmission rule.

Algorithm 2 Mem-Spike Output Generation

Require: Membrane potential $U[t]$, spike signal $S[t]$, threshold v_{th}
Ensure: Output $O[t]$
if $U[t] \geq v_{th}$ **then**
    $S[t] \leftarrow 1$
else
    $S[t] \leftarrow 0$
$U[t] \leftarrow U[t] - S[t] \times v_{th}$
$O[t] \leftarrow S[t] \times U[t]$
return $O[t]$

4.4. Residual-Membrane Connection

As deep learning models become deeper, their performance can degrade due to the vanishing gradient problem during training, and residual learning has been widely adopted as a representative technique to mitigate this issue [2]. Residual blocks employ skip connections that directly link the output of a layer to a subsequent layer, reducing information loss and improving training stability. However, such mechanisms are typically implemented by forwarding post-activation outputs, which is fundamentally different from SNNs that rely on discrete spike events and membrane-potential dynamics.

In conventional SNNs, inter-layer information transmission is easily constrained by spike sparsity, which can lead to degraded signal propagation and unstable learning as the network becomes deeper. To alleviate this issue, we propose a Residual-Membrane Connection that adapts the concept of residual learning to the spike-based and membrane-potential dynamics of SNNs. Unlike conventional skip connections that forward a neuron's activation output, our approach forwards the input-layer membrane potential as a residual term to each hidden layer, thereby compensating for information loss.

In the proposed mechanism, the membrane potential of the l-th layer at time t is defined as:

$$U^{(l)}[t] = \beta U^{(l)}[t-1] + I^{(l)}[t] - S^{(l)}[t-1]v_{th} + \gamma \cdot U^{(0)}[t],$$

where $U^{(l)}[t]$ is the membrane potential at layer l, β is the leak coefficient, $I^{(l)}[t]$ is the input current, $S^{(l)}[t-1]$ indicates whether a spike occurred at the previous step, and γ controls the contribution of the residual term. Here, $U^{(0)}[t]$ denotes the input-layer membrane state computed from the encoded input at time t. This mechanism differs from conventional residual connections in that it propagates membrane potentials—rather than

https://doi.org/10.3390/electronics15081597

output spikes—through skip connections, thereby mitigating signal vanishing caused by spike sparsity.

Residual-Membrane Connection is designed to preserve input information while remaining compatible with the accumulative dynamics of SNNs. This is particularly suitable for problems with strong temporal dependencies, such as time-series forecasting, where the input-layer membrane potential can continuously influence higher-layer activations. By providing richer contextual information, the proposed mechanism can improve prediction stability in our setting. Overall, it is designed to facilitate information flow and stable learning in deep SNNs, helping address limitations of conventional SNN architectures. The overall Residual-Membrane update is summarized in Algorithm 3.

Algorithm 3 Residual-Membrane Connection

Require: $U[t-1]$, $I[t]$, $S[t-1]$, $U_{\text{input}}[t]$, β, v_{th}, γ
Ensure: $U[t]$
1: $U[t] \leftarrow \beta\, U[t-1] + I[t] - S[t-1]\, v_{th} + \gamma\, U_{\text{input}}[t]$
2: **return** $U[t]$

5. Experiments and Results

This section conducts experiments to validate the effectiveness of the proposed Delta-Rate Encoding, Mem-Spike Mechanism, and Residual-Membrane Connection, and their performance is evaluated through a range of comparative studies. We use representative time-series forecasting benchmark datasets and compare against conventional SNN baselines and alternative encoding strategies to demonstrate the impact of the proposed mechanisms. In addition, we analyze layer-wise spike activity to further examine the efficiency characteristics of SNNs.

5.1. Experimental Setup

To evaluate electricity load forecasting performance, we use two public datasets, *Energy* and *Electricity*. The former corresponds to the Almanac of Minutely Power Dataset (AMPds2) [52], which we aggregate to an hourly resolution using hourly summation. The latter refers to the public Electricity Load Diagrams 2011–2014 dataset [53]. To improve reproducibility, we explicitly define the forecasting setting for each dataset. For the AMPds-based Energy dataset, we predict the WHE kWh series using 18 input features: *date*, *hour*, *kWh*, *temp*, *hum*, *pres*, *hol*, *day*, *month*, *week*, *day_hour_mean*, *hour_mean*, *hour_std*, *sin_time*, *cos_time*, *THI*, *CDH*, and *HDH*. Thus, the Energy dataset is used for single-target multi-step forecasting with multivariate input, where $X \in \mathbb{R}^{B\times W\times 18}$ and $Y \in \mathbb{R}^{B\times H}$ for $H \in \{6, 24, 48\}$. For the Electricity dataset, each timestep is represented by a 321-dimensional consumption vector obtained by row-wise parsing, linear interpolation, and Min-Max normalization. These 321 values are used as multivariate input features, while a single target kWh series (`[TARGET COLUMN/INDEX]`) is predicted. Accordingly, the Electricity dataset is also formulated as a single-target multi-step forecasting task with multivariate input, where $X \in \mathbb{R}^{B\times W\times 321}$ and $Y \in \mathbb{R}^{B\times H}$ for $H \in \{6, 24, 48\}$. For each dataset, we first split the raw time series chronologically and then generate sliding-window samples separately within each split to avoid temporal leakage. The training portion is further divided into training and validation subsets for early stopping and model selection. For each dataset horizon setting, training was repeated with five different random seeds, and the final reported result corresponds to the run with the best validation performance.

We adopt a direct forecasting setting by training one model per horizon, with horizons defined as $h \in \{6, 24, 48\}$. For each horizon, we set the input window length as

https://doi.org/10.3390/electronics15081597

$(W, h) \in \{(24, 6), (96, 24), (168, 48)\}$. Since both datasets are hourly, each horizon corresponds to 6/24/48 h ahead.

We consider both deep learning and SNN baselines. Specifically, we evaluate: (i) a deep learning model (LSTM), (ii) conventional SNN models (LIF, RLIF), and (iii) enhanced SNN models incorporating the proposed mechanisms (Proposed-LIF, Proposed-RLIF). All models are implemented using PyTorch (v2.7.0) and snnTorch (v0.9.4) [54].

We use mean squared error (MSE) as the loss function and optimize models with NAdam (initial learning rate 0.001). We set the batch size to 64 and train for up to 500 epochs with early stopping. To mitigate training instability caused by the non-differentiability of spikes, we apply surrogate-gradient backpropagation using a fast-sigmoid surrogate. For normalization, we apply feature-wise MinMax scaling fitted on the training set and reuse the same scaler for the validation and test sets.

For Delta-Rate encoding, we fix the threshold to 0.5, set the number of timesteps to $T = 100$, and use `off_spike = true`. All comparisons are conducted under the same preprocessing and training settings for each dataset (Energy and Electricity) and horizon (6/24/48), and the reported results are computed on the test set. In the experimental setting of this study, a threshold of 0.5 empirically provided the best performance and was therefore fixed across datasets and forecasting horizons to ensure a controlled and directly comparable evaluation. However, this value should not be regarded as universally optimal, since the most suitable threshold may vary depending on the dataset, task characteristics, and noise conditions. Adaptive or data-driven thresholding strategies therefore remain an important direction for future work.

5.2. Evaluation Metrics

We use R^2 and relative squared error (RSE) for performance comparison. R^2 measures how well the model explains the variance of the observations, where higher values indicate better performance. We define RSE as:

$$RSE = \sqrt{\frac{\sum_t (y_t - \hat{y}_t)^2}{\sum_t (y_t - \bar{y})^2}}.$$

To quantify spike activity, we report a normalized layer-wise firing rate rather than a raw count. For a spike tensor $\texttt{spk} \in \{0, 1\}^{B \times T \times N}$, we compute $\sum \texttt{spk} / (B \cdot T \cdot N)$ over the test set, where T denotes the number of timesteps used to generate spike sequences. In our experiments, T is fixed to 100.

5.3. Experimental Results

5.3.1. Forecasting Performance

Table 1 compares forecasting performance across baselines (LSTM, LIF, and RLIF) and the proposed SNN enhancements (Proposed-LIF and Proposed-RLIF) on Energy and Electricity for horizons $h \in \{6, 24, 48\}$. Overall, the proposed models improve the coefficient of determination (R^2) while reducing RSE compared to their corresponding baseline SNNs.

Proposed-LIF on the Energy dataset reduces error at longer horizons. At $h = 48$, RSE decreases from 0.4630 to 0.2873 (a 37.9% reduction), while R^2 improves from 0.5980 to 0.7635. On Electricity, Proposed-LIF shows a pronounced improvement at $h = 24$: RSE drops from 0.7770 to 0.4009 (a 48.4% reduction), and the negative R^2 of the baseline LIF (-0.0037) becomes positive (0.7467).

Proposed-RLIF yields particularly large gains on Energy. At $h = 48$, RSE decreases from 0.5000 to 0.1922 (a 61.6% reduction) and R^2 increases from 0.5720 to 0.8599, suggesting that the proposed framework effectively mitigates RLIF's representational limitations in long-horizon forecasting. On the Electricity dataset, Proposed-RLIF also improves over

https://doi.org/10.3390/electronics15081597

RLIF; for example, at $h = 24$, R^2 increases from 0.6590 to 0.8152 and RSE decreases from 0.3870 to 0.3389 (a 12.4% reduction).

Table 1. Forecasting results on Energy and Electricity dataset for $h \in \{6, 24, 48\}$. DNN(LSTM), Baseline SNN(LIF/RLIF), and Proposed-LIF/RLIF are compared using R^2 and RSE.

Model	Metric	Energy Dataset			Electricity Dataset		
		6	**24**	**48**	**6**	**24**	**48**
LSTM	R^2	0.8927	0.8760	0.8200	0.8883	0.8733	0.8659
	RSE	0.1822	0.1746	0.2233	0.2562	0.2750	0.2851
LIF	R^2	0.7788	0.6836	0.5980	0.5910	−0.0037	−0.0055
	RSE	0.2812	0.3754	0.4630	0.7138	0.7770	0.7791
Proposed-LIF	R^2	0.8219	0.7916	0.7635	0.8090	0.7467	0.7184
	RSE	0.2462	0.2774	0.2873	0.3447	0.4009	0.4210
RLIF	R^2	0.6923	0.6565	0.5720	0.6940	0.6590	0.6277
	RSE	0.3385	0.3849	0.5000	0.3373	0.3870	0.4470
Proposed-RLIF	R^2	0.8791	0.8716	0.8599	0.8599	0.8152	0.7456
	RSE	0.1871	0.2087	0.1922	0.2916	0.3389	0.4011

We also observe negative R^2 values for LIF on Electricity at $h \in \{24, 48\}$, indicating performance worse than a naive predictor that always outputs the target mean. This behavior is consistent with the limited expressiveness of simple spike-based dynamics under increasing long-range dependency and accumulated forecasting errors. In contrast, applying the proposed mechanisms mitigates this failure mode, as evidenced by the simultaneous improvements in both R^2 and RSE for Proposed-LIF.

5.3.2. Layer-Wise Spike Firing Rates

To assess how spiking activity and predictive performance change as the network becomes deeper, we report layer-wise spike firing rates together with forecasting metrics. We define the normalized firing rate as $r = \sum \text{spk}/(B \cdot T \cdot N)$, computed over the entire test set, where B is the batch dimension, T is the number of encoding steps (set to $T = 100$ in our experiments), and N denotes the number of neurons in the corresponding layer.

Table 2 (Electricity, $h = 6$) shows that increasing the depth from 2 to 5 layers can severely degrade the baseline RLIF, even though the firing rates in the deeper RLIF remain relatively high. Specifically, RLIF at depth 5 yields a near-zero (slightly negative) R^2 (-0.0004) with a substantially larger error (RSE $= 0.8181$), whereas RLIF at depth 2 achieves $R^2 = 0.8085$ and RSE $= 0.3434$. Note that $R^2 < 0$ indicates performance worse than a simple mean predictor.

Table 2. Layer-wise spike firing rates and forecasting performance (Electricity, $h = 6$). "Spk#_rate" indicates an average firing rate at layer #.

Model	Depth	Spk1_Rate	Spk2_Rate	Spk3_Rate	Spk4_Rate	Spk5_Rate	R^2	RSE
RLIF	2	0.4635	0.4668	–	–	–	0.8085	0.3434
	5	0.5851	0.6563	0.4867	0.5000	0.4375	−0.0004	0.8181
Proposed-RLIF	2	0.4329	0.2332	–	–	–	0.8147	0.3373
	5	0.3652	0.3866	0.4429	0.4289	0.3664	0.8599	0.2916

In contrast, Proposed-RLIF maintains strong forecasting performance at depth 5 ($R^2 = 0.8599$, RSE $= 0.2916$) while exhibiting overall lower firing rates than the baseline RLIF at the same depth. This suggests that, in our evaluation setting, the proposed

https://doi.org/10.3390/electronics15081597

framework can improve spike-mediated communication: it achieves comparable or better predictions with fewer spikes, rather than relying on increased spiking activity. Overall, these results are consistent with the design goal of enhancing information preservation and transmission in deep SNNs through improved encoding and membrane-state-based communication mechanisms. This trend is expected because the proposed mechanisms increase the information carried per spike and stabilize inter-layer signal propagation. Specifically, Delta–Rate Encoding captures subthreshold variations via accumulation, reducing the need for frequent firing to represent smooth changes, while Mem–Spike transmits a real-valued membrane payload at firing times, improving spike efficiency. In addition, Residual–Membrane connections alleviate depth-induced information vanishing, allowing accurate prediction with more balanced—and often lower—spiking activity.

5.3.3. Ablation Test

To assess the contribution of each mechanism, we conduct an ablation study on the Electricity dataset with $h = 6$ using the 5–layer Proposed-RLIF model. This ablation study was conducted on the best-performing configuration in our experiments and is intended as a representative diagnostic analysis rather than a comprehensive sweep across all datasets, horizons, and depths. Because the proposed components interact within the full model, the observed changes should be interpreted as configuration-dependent removal effects in this setting, rather than as exact and universally separable contributions. We remove Delta–Rate Encoding, Mem–Spike, and Residual–Membrane one at a time and re-evaluate forecasting performance. As shown in Table 3, removing Delta–Rate Encoding reduces R^2 to 0.7643 and increases RSE to 0.3833, removing Mem–Spike yields R^2 = 0.7818 and RSE = 0.3679, and removing Residual–Membrane results in the largest degradation (R^2 = 0.7564, RSE = 0.3875). Overall, compared with the full model, ablations reduce R^2 by approximately 5.6–8.7% and increase RSE by approximately 13.8–19.9%.

Table 3. An ablation test was conducted on the Proposed_RLIF model, and the results are reported in terms of R^2 and RSE. DR_Encoding indicates that the mechanisms were removed.

Model (Ablation)	Metric	None	DR_Encoding	Mem-Spike	Res-Mem
Proposed_RLIF	R^2	0.8599	0.7643	0.7818	0.7564
	RSE	0.2916	0.3833	0.3679	0.3875

The main takeaway of this ablation is qualitative rather than universal in magnitude. The results indicate that the proposed mechanisms are jointly useful in the tested setting, but the exact contribution of each mechanism cannot be uniquely isolated and may vary across datasets, horizons, and model configurations.

5.3.4. Energy Efficiency

Mem–Spike transmits the membrane potential as a real-valued payload at firing times, improving accuracy but increasing computation and memory overhead relative to a purely binary event-driven pathway. Mem–Spike transmits the membrane potential as a real-valued payload at firing times, thereby improving accuracy at the cost of additional computational and memory overhead relative to a purely binary event-driven pathway. Because neuromorphic hardware is not available in our setting, we do not report directly measured device power. Instead, following the estimation methodology in [55], we report operation-level energy estimates obtained by counting the dominant forward-pass operations and applying the same assumptions to all compared models. Under this common estimation methodology and evaluation setting, the conventional DNN is estimated at 358.141 μJ, Proposed-RLIF at 32.69 μJ, and baseline RLIF at 7.09 μJ. Accordingly, the

https://doi.org/10.3390/electronics15081597

proposed method is more efficient than the selected deep learning baseline under the same estimation methodology, but more costly than baseline RLIF because Mem–Spike introduces real-valued membrane–payload transmission and its associated operations. This result should therefore be interpreted as an accuracy–efficiency trade-off under a common estimation methodology, rather than as a direct hardware power measurement. From a broader deployment perspective, system-level embedded studies such as the reviewer-recommended DOI paper further highlight that acquisition, communication, and platform-level constraints must also be taken into account; our analysis complements such work by focusing specifically on model-level inference cost.

6. Conclusions

This study proposed a new framework that combines *Delta–Rate Hybrid Encoding*, the *Mem–Spike Mechanism*, and the *Residual–Membrane Connection* to mitigate two major challenges that arise when using SNNs for forecasting: signal information loss and signal sparsity (caused by low spike activity). The proposed encoding preserves fine-grained input variations by capturing abrupt changes via Delta encoding and accumulating subthreshold changes via Rate encoding. Mem–Spike reduces the loss inherent to spike-based representations by augmenting spike events with membrane-potential information at firing times, while Residual–Membrane alleviates information vanishing in deep SNNs to promote more persistent signal propagation across depth.

Experimental results show that the proposed SNN enhancements (Proposed–LIF and Proposed–RLIF) consistently improve upon baseline SNNs (LIF and RLIF) across two datasets and multiple forecasting horizons, and in some settings achieve performance that approaches the deep learning baseline (LSTM). Layer-wise firing analysis further indicates that the proposed configuration exhibits more balanced firing activity and more stable signal propagation in regimes where baseline models degrade. Ablation studies also support that the three mechanisms contribute in a complementary manner, where removing each mechanism leads to reduced explanatory power and increased error.

Nevertheless, the intrinsic limitation of spike-based representations remains: information is not immediately transmitted at non-firing times. When no spike is emitted, the output is zero; that is, information is not conveyed at every time step. Subthreshold variations are reflected only after accumulation, which can blur fine-grained fluctuations in real time. Moreover, converting continuous signals into binary spikes is inherently non-invertible, making some information loss unavoidable. The current evaluation is limited to two energy/load datasets, so broader domain generalization to noisier or biomedical time-series settings remains to be validated. We also did not conduct a systematic scalability analysis on substantially deeper architectures, and a dedicated interpretability analysis was outside the scope of this work. Accordingly, validation on deeper networks and integration with explanation-oriented analysis tools remain important directions for future research. This may partly explain why our method does not surpass the selected deep learning baseline (LSTM) in absolute accuracy. While it substantially improves baseline SNN forecasting and narrows the gap to that baseline, fully matching continuous-valued DNNs remains challenging because we focus on conventional LIF/RLIF-based SNNs. Future work will further explore designs that maintain information transfer capability at low firing activity.

Author Contributions: Conceptualization, J.L. and E.-K.L.; methodology, J.L.; software, J.L.; validation, J.L. and E.-K.L.; formal analysis, J.L.; investigation, J.L. and E.-K.L.; resources, J.L. and E.-K.L.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and E.-K.L.; visualization, J.L.; supervision, E.-K.L.; project administration, E.-K.L.; and funding acquisition, J.L. and E.-K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Incheon National University Research Grant in 2025.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in [AMPds2] at https://doi.org/10.7910/DVN/FIE0S4 (accessed on 7 April 2026) and in [Electricity Load Diagrams 2011–2014] at https://doi.org/10.24432/C58C86 (accessed on 7 April 2026).

Acknowledgments: During the preparation of this manuscript, the author(s) used ChatGPT 5.2 for the purposes of editing grammatical errors and expressions. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Lake Tahoe, NV, USA, 3–6 December 2012.
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2016.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 3–6 December 2017.
4. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the NAACL-HLT*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019.
5. Pathak, J.; Subramanian, S.; Harrington, P.; Raja, S.; Chattopadhyay, A.; Mardani, M.; Kurth, T.; Hall, D.; Li, Z.; Azizzadenesheli, K.; et al. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators. *arXiv* **2022**, arXiv:2202.11214.
6. Ryu, S.; Noh, J.; Kim, H. Deep Neural Network Based Demand Side Short Term Load Forecasting. *Energies* **2017**, *10*, 3. [CrossRef]
7. Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Ranzato, M.A.; Senior, A.; Tucker, P.; Yang, K.; et al. Large Scale Distributed Deep Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Lake Tahoe, NV, USA, 3–6 December 2012.
8. Strubell, E.; Ganesh, A.; McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019.
9. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
10. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv* **2016**, arXiv:1510.00149. [CrossRef]
11. Lai, G.; Chang, W.-C.; Yang, Y.; Liu, H. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *Proceedings of the SIGIR*; Association for Computing Machinery: New York, NY, USA, 2018.
12. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [CrossRef]
13. Indiveri, G.; Liu, S.-C. Memory and Energy Efficient Neuromorphic Systems. *Proc. IEEE* **2015**, *103*, 1379–1397. [CrossRef]
14. Furber, S.B. Large-scale Neuromorphic Computing Systems. *J. Neural Eng.* **2016**, *13*, 051001. [CrossRef]
15. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface. *Science* **2014**, *345*, 668–673. [CrossRef] [PubMed]
16. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* **2018**, *38*, 82–99. [CrossRef]
17. Furber, S.B.; Galluppi, F.; Temple, S.; Plana, L.A. The SpiNNaker Project. *Proc. IEEE* **2014**, *102*, 652–665. [CrossRef]
18. Maass, W. Networks of Spiking Neurons: The Third Generation of Neural Network Models. *Neural Netw.* **1997**, *10*, 1659–1671. [CrossRef]
19. Gerstner, W.; Kistler, W.M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*; Cambridge University Press: Cambridge, UK, 2002.
20. Tavanaei, A.; Ghodrati, M.; Kheradpisheh, S.R.; Masquelier, T.; Maida, A. Deep Learning in Spiking Neural Networks. *Neural Netw.* **2019**, *111*, 47–63. [CrossRef]

21. Pfeiffer, M.; Pfeil, T. Deep Learning with Spiking Neurons: Opportunities and Challenges. *Front. Neurosci.* **2018**, *12*, 774. [CrossRef] [PubMed]
22. Roy, K.; Jaiswal, A.; Panda, P. Towards Spike-Based Machine Intelligence with Neuromorphic Computing. *Nature* **2019**, *575*, 607–617. [CrossRef]
23. Rueckauer, B.; Lungu, I.-A.; Hu, Y.; Pfeiffer, M.; Liu, S.-C. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. *Front. Neurosci.* **2017**, *11*, 682. [CrossRef] [PubMed]
24. Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Shi, L. Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks. *Front. Neurosci.* **2018**, *12*, 331. [CrossRef] [PubMed]
25. Shrestha, S.B.; Orchard, G. SLAYER: Spike Layer Error Reassignment in Time. *arXiv* **2018**, arXiv:1810.08646. [CrossRef]
26. Yılmaz, E.; Gevrek, Ö.B.; Wu, J.; Chen, Y.; Meng, X.; Li, H. Deep Convolutional Spiking Neural Networks for Keyword Spotting. *Proc. Interspeech* **2020**, 2557–2561. [CrossRef]
27. Auge, D.; Hille, J.; Mueller, E.; Knoll, A. A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks. *Neural Process. Lett.* **2021**, *53*, 4693–4710. [CrossRef]
28. Kim, J.; Kim, H.; Huh, S.; Lee, J.; Choi, K. Deep Neural Networks with Weighted Spikes. *Neurocomputing* **2018**, *311*, 373–386. [CrossRef]
29. Bellec, G.; Salaj, D.; Subramoney, A.; Legenstein, R.; Maass, W. Long short-term memory and learning-to-learn in networks of spiking neurons. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 3–8 December 2018.
30. Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; Yuan, L. Spikformer: When Spiking Neural Network Meets Transformer. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
31. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 3rd ed.; OTexts: Melbourne, Australia, 2021.
32. Hong, T.; Fan, S. Probabilistic electric load forecasting: A tutorial review. *Int. J. Forecast.* **2016**, *32*, 914–938. [CrossRef]
33. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*, 5th ed.; Wiley: Hoboken, NJ, USA, 2015.
34. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
35. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
36. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
37. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the EMNLP, Doha, Qatar, 25–29 October 2014.
38. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
39. Zhao, X.; Peng, H.; Zhang, L.; Ma, H. Research on a Short-Term Power Load Forecasting Method Based on a Three-Channel LSTM-CNN Hybrid Model. *Electronics* **2025**, *14*, 2262.
40. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual Event, 6–14 December 2021.
41. Liu, Y.; Wu, H.; Wang, J.; Long, M. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. *arXiv* **2024** arXiv:2310.06625. [CrossRef]
42. Lv, C.; Wang, Y.; Han, D.; Zheng, X.; Huang, X.; Li, D. Efficient and Effective Time-Series Forecasting with Spiking Neural Networks. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*; PMLR: Cambridge, MA, USA, 2024.
43. Feng, S.; Feng, W.; Gao, X.; Zhao, P.; Shen, Z. TS-LIF: A Temporal Segment Spiking Neuron Network for Time Series Forecasting. 2025. Available online: https://openreview.net/forum?id=rDe9yQQYKt (accessed on 7 April 2026).
44. Hu, B.; Wang, Y.; Lu, M.; Chang, Z. SpikeSTAG: Spatial-Temporal Forecasting via GNN-SNN Collaboration. *arXiv* **2025**, arXiv:2508.02069.
45. Diehl, P.U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.-C.; Pfeiffer, M. Fast-classifying, High-accuracy Spiking Deep Networks Through Weight and Threshold Balancing. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2015. [CrossRef]
46. Hu, Y.; Zheng, Q.; Jiang, X.; Pan, G. Fast-SNN: Fast Spiking Neural Network by Converting Quantized ANN. *arXiv* **2023**, arXiv:2305.19868. [CrossRef] [PubMed]
47. Li, Y.; Deng, S.; Dong, X.; Gong, R.; Gu, S. A Free Lunch From ANN: Towards Efficient, Accurate Spiking Neural Networks Calibration. *arXiv* **2021**, arXiv:2106.06984. [CrossRef]
48. Neftci, E.O.; Mostafa, H.; Zenke, F. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based Optimization to Spiking Neural Networks. *IEEE Signal Process. Mag.* **2019**, *36*, 51–63. [CrossRef]
49. Han, B.; Srinivasan, G.; Roy, K. RMP-SNN: Residual Membrane Potential Neuron for Enabling Deeper High-Accuracy and Low-Latency Spiking Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2020.

50. Sengupta, A.; Ye, Y.; Wang, R.; Liu, C.; Roy, K. Going Deeper in Spiking Neural Networks: VGG and Residual Architectures. *Front. Neurosci.* **2019**, *13*, 95. [CrossRef]
51. Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; Tian, Y. Deep Residual Learning in Spiking Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual Event, 6–14 December 2021.
52. Makonin, S.; Ellert, B.; Bajić, I.V.; Popowich, F. Electricity, water, and natural gas consumption of a residential house in Canada from 2012 to 2014 (AMPds2). *Sci. Data* **2016**, *3*, 160037. [CrossRef]
53. Trindade, A. Electricity Load Diagrams 2011–2014 Data Set. *UCI Mach. Learn. Repos.* **2015**. [CrossRef]
54. Eshraghian, J.K.; Ward, M.; Neftci, E.O.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D.S.; Lu, W.D. Training Spiking Neural Networks Using Lessons from Deep Learning. *Proc. IEEE* **2023**, *111*, 1016–1054. [CrossRef]
55. Rathi, N.; Roy, K. DIET-SNN: A Low-Latency Spiking Neural Network with Direct Input Encoding and Leakage and Threshold Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3174–3182. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Ensemble Graph Neural Networks for Probabilistic Sea Surface Temperature Forecasting via Input Perturbations

Alejandro J. González-Santana [1], Giovanny A. Cuervo-Londoño [2] and Javier Sánchez [1,*]

[1] Centro de Tecnologías de la Imagen, Instituto Universitario de Cibernética, Empresas y Sociedad (IUCES), Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas, Spain; alejandro.gonzalez147@alu.ulpgc.es
[2] Oceanografía Física y Geofísica Aplicada, Instituto Universitario en Acuicultura Sostenible y Ecosistemas Marinos (ECOAQUA), Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas, Spain; giovanny.cuervo@ulpgc.es
* Correspondence: jsanchez@ulpgc.es; Tel.: +34-928-458710

Abstract

Accurate regional ocean forecasting requires models that are both computationally efficient and capable of representing predictive uncertainty. This work investigates ensemble learning strategies for sea surface temperature (SST) forecasting using Graph Neural Networks (GNNs), with a focus on how input perturbation design affects forecast skill and uncertainty representation. We adapt a GNN architecture to the Canary Islands region in the North Atlantic and implement a homogeneous ensemble approach inspired by bagging, where diversity is introduced during inference by perturbing initial ocean states rather than retraining multiple models. Several noise-based ensemble generation strategies are evaluated, including Gaussian noise, Perlin noise, and fractal Perlin noise, with systematic variation of noise intensity and spatial structure. Ensemble forecasts are assessed over a 15-day horizon using deterministic metrics (RMSE and bias) and probabilistic metrics, including the Continuous Ranked Probability Score (CRPS) and the Spread–skill ratio. The results show that, while deterministic skill remains comparable to the single-model forecast, the type and structure of input perturbations influence uncertainty representation, particularly at longer lead times. Ensembles generated with spatially coherent perturbations, such as low-resolution Perlin noise, achieve improved calibration and lower CRPS compared to purely random Gaussian perturbations. These findings highlight the role of noise structure and scale in ensemble GNN design, indicating that specifically structured input perturbations can improve ensemble diversity and calibration without additional training cost. These results provide a methodological contribution toward the study of ensemble-based GNN approaches for regional ocean forecasting.

Keywords: graph neural network; ensemble learning; sea surface temperature; probabilistic ocean forecasting; structured noise perturbations

1. Introduction

The increasing relevance of the blue economy [1], the accelerating impacts of climate change, and the objectives defined within the Sustainable Development Goals have intensified the demand for accurate and timely ocean forecasting systems [2]. Reliable predictions of oceanographic variables, such as sea surface temperature (SST), are essential for maritime operations, ecosystem monitoring, fisheries management, and climate-related decision-making. Traditionally, these forecasts have relied on numerical ocean models that explicitly solve the physical equations governing ocean dynamics. While physically

grounded and robust, such models are computationally expensive and often restricted to large operational centers, limiting their accessibility and adaptability to regional and high-resolution applications [3].

Recent advances in machine learning (ML) have introduced an alternative paradigm for geophysical prediction, enabling data-driven models to learn complex spatiotemporal relationships directly from observations and reanalysis products [4]. Leveraging modern hardware accelerators, ML-based models offer orders-of-magnitude faster inference compared to numerical solvers, making them particularly attractive for operational and near-real-time forecasting. Although many state-of-the-art ML climate models operate at global scales, recent work has demonstrated their growing potential for regional applications, where higher spatial resolution is required to capture localized ocean processes such as coastal upwelling and mesoscale variability.

Within this context, Graph Neural Networks (GNNs) have emerged as a powerful framework for modeling geophysical systems defined over irregular spatial domains [5]. By representing spatial locations as nodes and their physical relationships as edges, GNNs naturally handle complex geometries such as coastlines and bathymetry, which are difficult to model using regular grids. Architectures such as GraphCast [6] and its regional adaptations [7] have demonstrated that GNNs can achieve competitive or superior performance to traditional numerical models in medium-range forecasting while maintaining high computational efficiency. Recent works [8,9] extend this approach to regional ocean prediction through a hierarchical encoder–processor–decoder GNN tailored to oceanographic data.

Despite these advances, a major limitation of most deep learning-based forecasting systems is their deterministic nature, which hampers their ability to represent forecast uncertainty—an essential component for operational oceanography and climate services [10]. Ensemble forecasting [11] is the standard approach for uncertainty quantification in numerical weather and ocean prediction, typically achieved through perturbed initial conditions or stochastic parameterizations. Recently, ensemble methodologies have also been adopted in machine learning-based forecasting systems, including AIFS ENS [12], GenCast [13], NeuralGCM [14], and Pangu-Weather [15], demonstrating that ML ensembles can provide calibrated probabilistic forecasts at a fraction of the computational cost of traditional ensemble systems.

Ensemble learning relies fundamentally on diversity among its members: individual forecasts must differ sufficiently so that their errors are weakly correlated, allowing aggregation to improve robustness and reliability [16,17]. In the context of deep learning, diversity can be introduced through heterogeneous architectures, multiple independently trained models, or perturbations applied to inputs, parameters, or latent representations. However, training multiple high-capacity models is often computationally prohibitive, particularly for regional GNNs operating on high-resolution meshes.

This work explores a computationally efficient ensemble strategy for regional ocean forecasting using GNNs, implemented through input perturbation during inference rather than repeated training. Building on SeaCast [8], we adapt the model to the Canary Islands region in the North Atlantic, and investigate how different noise-based perturbation strategies affect ensemble performance. Specifically, we compare Gaussian noise and spatially structured perturbations based on Perlin and fractal Perlin noise, systematically analyzing the influence of noise intensity and spatial resolution.

The ensemble forecasts are evaluated using both deterministic metrics (RMSE and bias) and probabilistic metrics from WeatherBench [18], including the Continuous Ranked Probability Score (CRPS) and the spread–skill ratio. These metrics allow us to assess not only forecast accuracy but also ensemble calibration and uncertainty representation across multiple lead times. By focusing on inference-time perturbations and ensemble

https://doi.org/10.3390/electronics15081583

design choices, this study aims to clarify how uncertainty can be effectively represented in GNN-based regional ocean forecasts through lightweight ensemble construction.

In contrast to recent generative or multi-model ensemble approaches, this work focuses on lightweight ensemble construction aimed at reducing computational overhead in high-resolution GNN-based forecasting systems. Rather than retraining multiple networks or introducing stochasticity during training, we investigate how structured perturbations of the initial ocean state at inference time can induce forecast diversity while relying on a single trained model. This design enables ensemble generation from a single model and serves as a proof-of-concept for obtaining probabilistic information without the need for multiple independently trained models.

The main contributions of this work are threefold: First, we present an efficient ensemble framework for regional SST forecasting based on inference-time input perturbations applied to a hierarchical GNN; second, we provide a systematic comparison of unstructured (Gaussian) versus spatially coherent (Perlin and fractal Perlin) noise, showing that the latter is associated with more stable uncertainty estimates at longer lead times; third, we offer empirical guidance on noise scale and structure for ensemble GNN design in regional ocean applications.

The remainder of the paper is organized as follows: Section 2 reviews related work on ML-based geophysical forecasting and ensemble methods; Section 3 describes the dataset, GNN architecture, and ensemble generation strategies; Section 4 outlines the experimental setup and evaluation metrics; Section 5 presents the results; and Section 6 discuss the implications of our findings and conclude with directions for future research.

2. Related Work

Ensemble forecasting is central to uncertainty quantification in numerical weather prediction (NWP), where diversity is typically introduced through perturbed initial conditions and stochastic parameterizations. As ML models increasingly complement or replace traditional solvers in atmospheric and ocean forecasting, ensemble methodologies have been adapted to neural architectures [19] to provide probabilistic predictions. Recent work in ML-based geophysical forecasting can be organized into four principal ensemble paradigms: (i) independently trained multi-model ensembles, (ii) generative probabilistic models, (iii) hybrid physics–ML systems with stochastic components, and (iv) inference-time perturbation ensembles.

Operational AI systems such as AIFS ENS [12] follow the classical ensemble paradigm by training or fine-tuning multiple model instances. Diversity arises from independent optimization trajectories, initialization differences, or data subsampling, consistent with ensemble theory [16,20,21]. This strategy benefits from strong variance reduction and often yields well-calibrated forecasts. However, it requires substantial computational resources, limiting scalability for regional, high-resolution domains where training multiple large neural models is prohibitive.

A second paradigm is represented by diffusion-based systems such as GenCast [13]. These models learn the conditional distribution of future states and directly generate multiple coherent forecast trajectories. By optimizing distributional objectives, they achieve competitive CRPS performance relative to operational ensemble systems. Nevertheless, generative approaches [22–24] introduce considerable training complexity, high computational costs, and reduced interpretability, as uncertainty is encoded implicitly in latent representations rather than explicitly through physically motivated perturbations.

NeuralGCM [14] exemplifies a hybrid strategy in which neural components are embedded within a physically inspired dynamical core. Stochastic perturbations are applied to learned tendencies, and probabilistic losses such as CRPS guide training. This approach

https://doi.org/10.3390/electronics15081583

enhances physical consistency and uncertainty representation but increases architectural complexity and coupling between model components, complicating adaptation to irregular regional meshes and ocean-specific geometries.

Recent work highlights the importance of spatially coherent perturbations. Pangu-Weather [15] demonstrated that structured Perlin noise [25] applied to initial states can generate meaningful ensemble diversity while preserving spatial smoothness. Compared to spatially independent Gaussian perturbations, structured noise better respects geophysical correlation scales, reducing artificial high-frequency artifacts. However, prior studies have primarily focused on global atmospheric forecasting and have not systematically examined how perturbation resolution and scale influence probabilistic skill, particularly in regional ocean contexts.

In parallel, GNNs [26,27] have emerged as powerful tools for geophysical prediction [5,28] on irregular domains [29,30]. Systems such as GraphCast [6] demonstrate that encoder–processor–decoder GNN architectures achieve state-of-the-art medium-range forecasting skill with high computational efficiency. Yet these models are predominantly deterministic, and ensemble strategies for GNN-based geophysical forecasting remain underexplored. Broader GNN ensemble approaches introduce diversity via architectural variation or subgraph sampling [31,32], but these strategies typically require retraining and are not tailored to probabilistic forecasting metrics.

Existing ensemble paradigms reveal a trade-off: generative and multi-model systems offer strong probabilistic calibration but incur high computational cost, while simple perturbation schemes are inexpensive but may lack physical realism. This work explores homogeneous ensembles for hierarchical GNNs [7,33] in regional sea surface temperature forecasting [8,9,34], introducing diversity exclusively at inference time through controlled perturbations of the initial ocean state. We provide a systematic comparison between unstructured Gaussian noise and spatially coherent Perlin-based perturbations, explicitly analyzing the impact of noise structure and resolution on CRPS and spread–skill ratio [18,35,36]. Our results suggest that spatial coherence plays a key role in achieving well-calibrated medium-range forecasts, while the effect of perturbation complexity appears to be secondary under the evaluated configurations.

By focusing on structured inference-time perturbations within a regional GNN framework, this study proposes a lightweight approach to probabilistic ocean forecasting, exploring the trade-off between deterministic neural models and more resource-intensive generative ensemble systems in a computationally efficient setting.

3. Methods and Data

3.1. Dataset

This study relies on three primary data types: oceanographic, atmospheric forcing, and bathymetry. Oceanographic data were obtained from the Copernicus Marine Service (CMEMS), specifically the European North-West Shelf/Iberia–Biscay–Irish Seas High-Resolution L4 Sea Surface Temperature Reprocessed product. This Level 4 reanalysis merges satellite observations with quality control and gap-filling, providing consistent daily SST data from 1 January 1982 to 31 December 2023, at $0.05° \times 0.05°$ resolution over the North Atlantic, Bay of Biscay, Irish Sea, and part of the western Mediterranean. The spatial domain is bounded by the coordinates listed in Table 1. SST is measured at 20 cm depth, with an accompanying error field (not used for model training, as this study focuses on evaluating ensemble uncertainty generated through perturbations rather than propagating observational uncertainty). Data were processed into daily matrices. Figure 1 shows the SST for 1 January 2018, in our domain.

https://doi.org/10.3390/electronics15081583

Table 1. Coordinates delimiting the area of study.

Parameter	Value
Maximum latitude	34.525°
Minimum latitude	19.55°
Maximum longitude	−5.975°
Minimum longitude	−20.97°

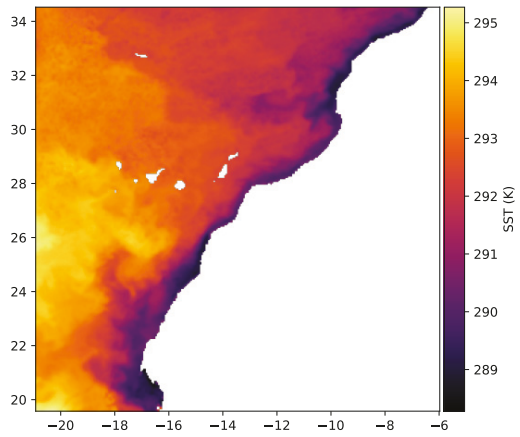


Figure 1. SST data in Kelvin for 1 January 2018, in the north-west African coast.

Atmospheric forcing data were obtained from the Copernicus Climate Data Store (C3S) ERA5 hourly data on single levels (1940–present). Two variables were selected: the east–west ($u10$) and north–south ($v10$) components of 10 m wind speed. Data were aggregated to daily means and interpolated to the CMEMS SST grid using bilinear interpolation, without the application of additional smoothing or filtering. The resulting fields were stored as daily arrays of ocean points. Figure 2 shows u and v winds for 1 January 2018.

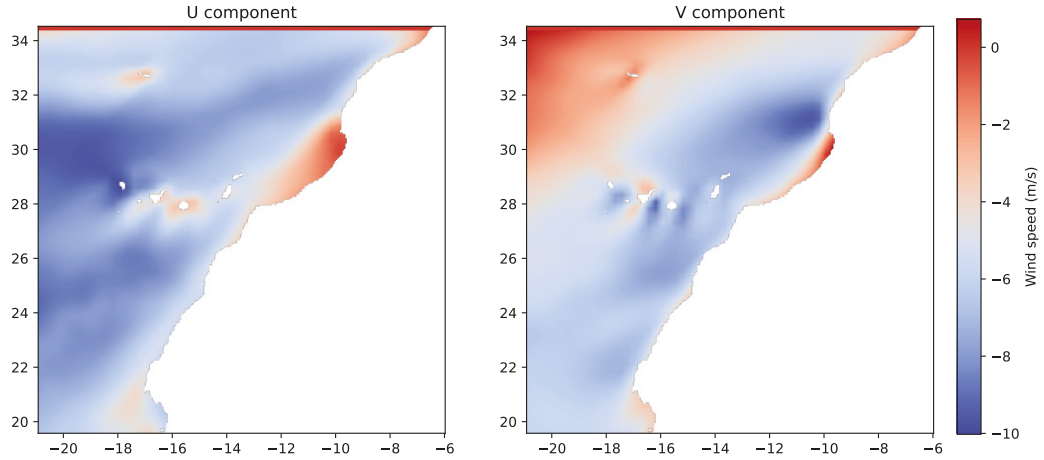


Figure 2. Wind components (u and v) at 10 m above the surface for 1 January 2018.

Bathymetry data were obtained from NOAA's ETOPO Global Relief Model 2022, providing ocean depth and land elevation at 0.0083° resolution. Depths were extracted for the study area, resampled to the SST grid, and land values were set to zero. Figure 3 shows the processed bathymetry.

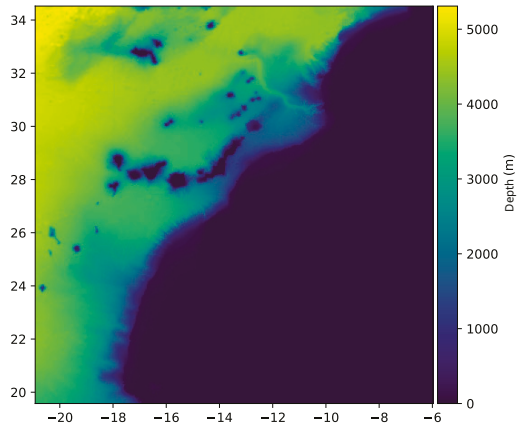


Figure 3. Processed bathymetry downloaded from ETOPO.

3.2. Graph Neural Network: Adapting to the North-Atlantic Subregion

SeaCast is a GNN designed for high-resolution, medium-term ocean predictions. The model leverages the complex geometry of the oceans through an autoregressive approach that combines historical oceanographic states, external forcings, and static information to predict future states. Originally developed for the Mediterranean Sea, it has been adapted in this study to the North Atlantic region.

It formalizes the prediction problem using temporal windows. Let the historical states of the oceanographic variables be denoted as

$$X^{-h:0} = (X^{-h}, \ldots, X^0),$$

where X^{-h} represents the state h days in the past and X^0 the current state. The future prediction window is

$$X^{1:T} = (X^1, \ldots, X^T),$$

where T denotes the prediction horizon in days. Forcing factors are defined analogously as

$$F^{1:T} = (F^1, \ldots, F^T).$$

The prediction relies on four components: initial and target states, forcings, and static data. The initial states consist of the previous day and the current value, while the target states correspond to the oceanographic variables over the T forecasted days. In this study, we set $T = 1$ for training and validation, and $T = 15$ for testing. During training, the model only requires one day of forcing information, as it is trained to predict one time step ahead. Thanks to its autoregressive design, predictions can be extended to any length T by sequentially generating states $\hat{X}^{1:T}$, starting from X^0. Figure 4 illustrates the autoregressive process.

The forecasting model used in this work was intentionally selected without autoregressive steps, as preliminary experiments showed only minor performance differences compared to a model trained with four autoregressive steps, while exhibiting substantially lower computational cost. A comparison between single-step and four-step autoregressive configurations is provided in Appendix A, confirming the small performance gap between both approaches.

The purpose of this work is not to optimize the forecasting model itself, but to analyze the effects of ensemble perturbation strategies on a fixed baseline. Using a single-step trained model allows us to isolate and study the impact of different noise configurations on prediction diversity without introducing additional variability from model retraining. However, this choice implies that the model is not specifically optimized for long autoregressive horizons, and therefore, results at longer lead times should be interpreted with caution.

https://doi.org/10.3390/electronics15081583

While this may accumulate errors in autoregressive prediction, as observed in medium-range weather forecasting systems such as Pangu-Weather [15], all compared ensemble configurations are evaluated under the same rollout procedure. Therefore, any bias induced by long-horizon propagation affects all methods equally, preserving the validity of relative comparisons between ensemble strategies.

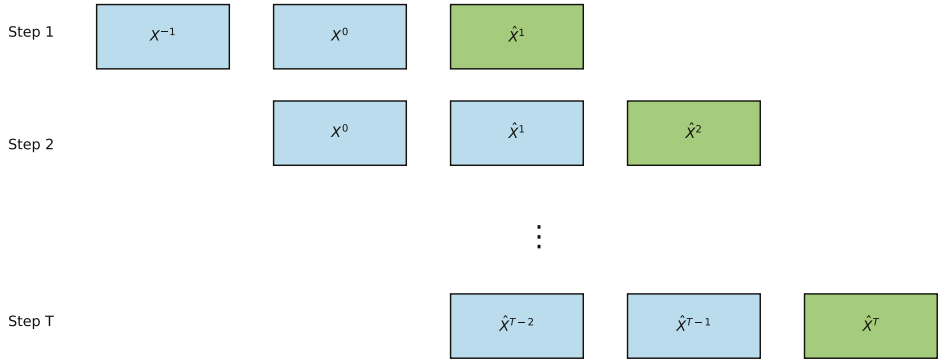


Figure 4. Diagram of autoregressive operation. States shown in blue correspond to the inputs used by the model to generate the next prediction, while states shown in green represent the predicted outputs.

In this work, SeaCast is used as a fixed baseline model, and no modifications are introduced to its core architecture. The changes applied are limited to domain adaptation to the North Atlantic region and adjustments required by computational constraints. Therefore, the focus of this study is placed on the analysis of ensemble perturbation strategies applied to a fixed forecasting backbone, rather than on model design.

SeaCast employs an encoder–processor–decoder architecture on a hierarchical mesh graph [7], where each stage uses an independent interaction network [37] to capture relationships between nodes and external influences. Inputs from the latitude–longitude grid are first projected onto the hierarchical mesh by the encoder, processed through sequential updates of latent representations in the processor, and finally mapped back to the grid by the decoder. The model predicts residual changes at each grid cell as:

$$\hat{X}_r^t = X_r^{t-1} + \mathrm{MLP}^{\mathrm{pred}}(v_r^G), \tag{1}$$

where v_r^G is the latent graph representation of cell r obtained from the mesh-to-grid connections. This residual formulation improves training stability and prediction accuracy while preserving the core GNN architecture.

To adapt the model to the North Atlantic, we made the following changes:

- Input constants were modified to accommodate new data sources, grid dimensions, and the number of oceanographic and atmospheric variables.
- The hierarchical mesh was adapted to the north-west African coast, including the Canary Islands.
- The mesh consists of three resolution levels (81, 27, 9 nodes per side), with the finest level including 81×81 nodes.
- Nodes and edges crossing land were removed, resulting in a total of 38,714 nodes and 49,061 grid-to-mesh connections.
- The number of neighbors per node in the hierarchy was reduced from 16 to 1 to improve computational efficiency.
- The processor includes four GNN layers, and hidden vector representations in the GNNs and MLPs have a dimension of 128.
- Training was performed on a single GPU with 32 GB RAM, requiring sequential processing of daily data and limiting batch size to one.

https://doi.org/10.3390/electronics15081583

- Autoregressive steps were initially increased from one to four, but later reduced to one in the final training to reduce computational cost.
- A noise module was introduced to generate diverse ensemble predictions.
- Evaluation tools from WeatherBench were modified to handle the custom data format.
- The final model included ∼5.6 million trainable parameters.

These modifications allow efficient operation in the North Atlantic while maintaining the core GNN architecture and autoregressive prediction capabilities. The model was trained to minimize the Mean Squared Error (MSE), considering the sequence of states generated by autoregressive steps, also known as rollout. The loss function is given by:

$$L = \frac{1}{T_{\text{rollout}}} \sum_{t=1}^{T_{\text{rollout}}} \sum_{i=1}^{C} \sum_{l=1}^{L_i} \frac{1}{|G_l|} \sum_{v \in G_{l(i)}} a_v \lambda_i \left(\hat{X}_{v,i}^t - X_{v,i}^t \right)^2, \tag{2}$$

where T_{rollout} is the number of autoregressive steps, C is the total number of variables, L_i is the number of depth levels of feature i, and G_l represents the number of grid points in the ocean at level l. The term a_v represents the weighting according to cell size, calculated as the cosine of latitude for each cell v and depth level l. Finally, λ_i is the inverse of the variance of the estimates for variable i [8].

3.3. Ensemble Method

Our ensemble learning strategy combines multiple models to generate a prediction during inference. This approach, inspired by the wisdom of the crowd , relies on the expectation that individual model errors will partially cancel out, yielding more accurate predictions than a single model. The outputs are combined using an aggregation function [20].

For an ensemble to outperform an individual model, base models must be better than random, and their errors should be as independent as possible [16]. Diversity among models is crucial: heterogeneous ensembles use different architectures (e.g., Stacking), while homogeneous ensembles use identical architectures with variations in inputs or parameters.

In this study, we adopted a bagging-inspired homogeneous ensemble strategy. Instead of retraining multiple models on bootstrapped datasets, a single GNN model is employed, and diversity is introduced at inference time by perturbing the oceanographic input data with noise. Predictions generated from perturbed initial states are aggregated through their mean to produce the final forecast. The types of noise and their characteristics are described in the following subsections.

It is important to note that this ensemble strategy primarily captures initial-condition uncertainty. All ensemble members share the same model parameters; therefore, model uncertainty associated with parameter estimation is not represented. The focus of this study is to evaluate how different types and configurations of input noise affect prediction diversity, rather than to construct a full ensemble covering all sources of uncertainty.

3.3.1. Gaussian

The simplest approach to introducing variability into the data is by adding random perturbations drawn from a normal distribution. Gaussian noise is widely used due to its prevalence in natural processes and favorable statistical properties. The distribution is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{3}$$

https://doi.org/10.3390/electronics15081583

where μ denotes the mean and σ^2 the variance. Gaussian noise has been employed in several state-of-the-art climate modeling frameworks included in performance benchmarks [36], such as ArchesWeatherGen [22], NeuralGCM [14], FuXi [38], and FourCastNet [39].

Figure 5 illustrates the effect of varying the standard deviation of the sampling distribution on noise intensity, assuming zero mean in all cases. A standard deviation of 1 yields perturbations that can reach values close to 4 in the distribution tails, whereas a standard deviation of 0.5 produces noticeably lower-amplitude noise due to reduced dispersion. When the standard deviation is set to 0.1, the perturbations become barely perceptible at the shared visualization scale. Despite these differences in magnitude, all realizations exhibit a spatially uncorrelated structure, resulting in uniformly scattered patterns across the domain.

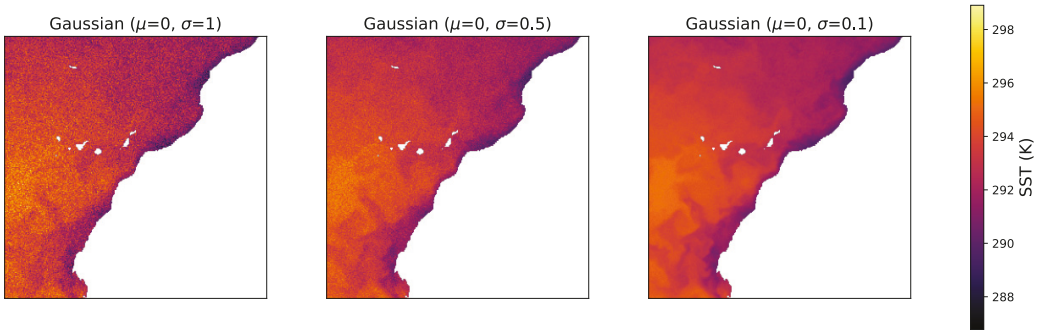


Figure 5. Examples of Gaussian noise added to SST, generated with different standard deviations.

3.3.2. Perlin Noise

In the context of ensemble construction, Perlin noise [25] offers a structured approach to promoting diversity while preserving spatial correlations. In contrast to Gaussian noise, which introduces spatially independent perturbations, Perlin noise generates smooth variations with controllable correlation lengths determined by the selected spatial resolution. This property enables the creation of ensemble members that differ in a physically consistent manner, maintaining large-scale coherence while introducing localized variability. Consequently, this type of noise facilitates diversification without injecting spurious high-frequency artifacts, making it particularly suitable for perturbing geophysical variables characterized by strong spatial dependencies. Perlin noise has been notably applied in Pangu-Weather [15] to generate random perturbations.

To generate the noise, a three-dimensional grid is first considered, whose points have integer coordinates. These points, which act as the vertices of the grid cubes, are assigned pseudo-random values and gradient vectors through a hash function H, which produces four mutually independent real values. For a point with coordinates (x, y, z), this assignment is represented by the following equation:

$$[a, b, c, d] = H(x, y, z), \tag{4}$$

where a, b and c represent the gradient and d the value of the function at the point (x, y, z).

If the evaluated point (x, y, z) coincides with a grid point, then the noise value equals d. Otherwise, if the point lies within the volume bounded by the vertices of a cube, the noise is computed through smooth interpolation (e.g., a cubic polynomial) using the values and gradients of the surrounding vertices. This interpolation is performed first along the x-axis (between edges), then along the y-axis (across cube faces), and finally along the z-axis (between planes). This procedure allows the generation of noise with a spatial component.

Figure 6 shows two different Perlin noise configurations that differ in the spatial resolution applied. In the case of higher spatial resolution on the latitude and longitude axes (2, 12, 12), a greater number of noise patterns can be observed, while in the low-

https://doi.org/10.3390/electronics15081583

resolution configuration (2, 3, 3), the patterns are larger and it is clearer to see how the interpolation smooths the transitions within each cell. Another important feature is that the higher resolution noise has a slightly higher intensity.

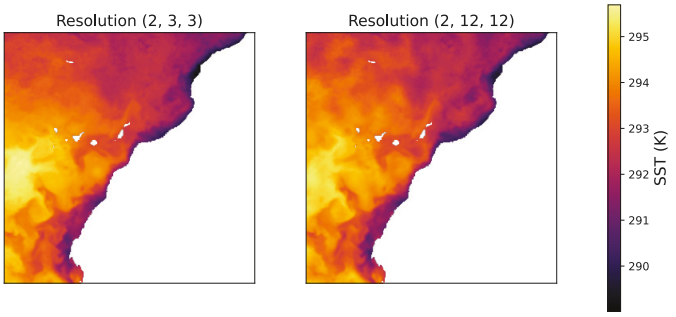


Figure 6. Examples of Perlin noise added to SST, generated with different spatial resolutions.

3.3.3. Fractal Perlin Noise

Another alternative for generating noise is the fractal Perlin noise. This function is distinguished by the use of three additional parameters. The first is octaves, which represent the number of iterations of the noise. Then there is persistence, which acts as the scaling factor for the noise amplitude between consecutive octaves. Finally, lacunarity is the factor of increase in frequency between two octaves. Its equation is given as:

$$F(x,y,z) = \alpha \cdot \sum_{i=0}^{O-1} a_i \cdot H(x,y,z;f_i), \tag{5}$$

where α represents the noise scale, O represents the number of octaves, a_i is the noise amplitude, and f_i is the frequency that modifies the resolution. The lacunarity, persistence, and resolution evolve as follows:

$$\text{resolution}_i = f_i \cdot \text{resolution}_0; \quad f_i = \text{lacunarity} \cdot f_{i-1}; \quad a_i = \text{persistence} \cdot a_{i-1}. \tag{6}$$

The noise resolution applied in each octave must always be an exact divisor of the dimensions of the desired noise shape.

Figure 7 shows two different configurations of fractal Perlin noise that differ only in the spatial resolution applied. Three octaves are used in the noise iteration, with a persistence of 0.5, which means that the intensity decreases progressively with each iteration. The lacunarity is two, which means that as the octaves progress, smaller and smaller patterns are used, resulting in more detailed and complex noise. In addition, a noise scalability coefficient ($\alpha = 0.2$) is applied to the noise resulting after the last octave.

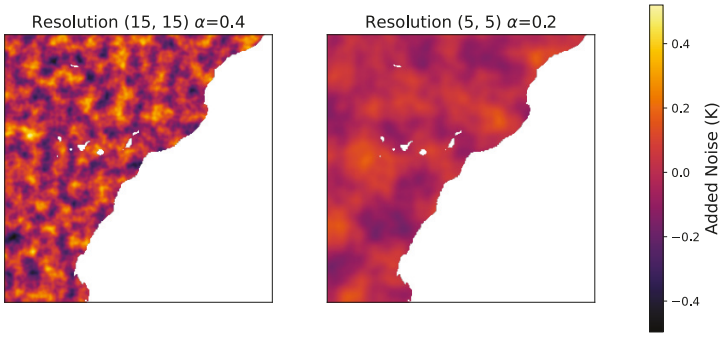


Figure 7. Examples of Perlin Fractal noise with three octaves added to the SST, generated with different spatial resolutions and the same noise scalability. The rest of the parameters correspond to default values.

https://doi.org/10.3390/electronics15081583

4. Experimental Setup

4.1. Evaluation Metrics

Two sets of metrics are used in this study: probabilistic metrics to evaluate the ensemble predictions generated from noisy inputs, and classical deterministic metrics, such as RMSE or Bias, to compare predictions against a single model without noise. This section focuses on the probabilistic metrics. Table 2 summarizes the notation used throughout the metric definitions.

Table 2. Notation used in the metrics [36].

Symbol	Rank	Description
f	-	Prediction
o	-	Observation
t	$1, \ldots, T$	Verification time
l	$1, \ldots, L$	Prediction horizon
i	$1, \ldots, I$	Latitude index
j	$1, \ldots, J$	Longitude index
m	$1, \ldots, M$	Set member index

The CRPS evaluates the accuracy of ensemble predictions, balancing the error against the internal dispersion of the set. For individual members X and observation Y, the CRPS is given by

$$\mathrm{CRPS}_{\mathrm{skill}} = \mathbb{E}|X - Y|, \tag{7}$$

$$\mathrm{CRPS}_{\mathrm{spread}} = \mathbb{E}[X - X'], \tag{8}$$

$$\mathrm{CRPS} = \mathrm{CRPS}_{\mathrm{skill}} - \frac{1}{2}\mathrm{CRPS}_{\mathrm{spread}} = \mathbb{E}|X - Y| - \frac{1}{2}\mathbb{E}|X - X'|. \tag{9}$$

The CRPS used in this study accounts for all set members and verification times [35] and is calculated as

$$\mathrm{CRPS}_l := \frac{1}{T}\sum_{t}^{T}\left(\frac{1}{M}\sum_{m=1}^{M}\left\|f^{(m)} - o\right\|_{t,l} - \frac{1}{2M(M-1)}\sum_{m=1}^{M}\sum_{\substack{n=1 \\ n\neq m}}^{M}\left\|f^{(m)} - f^{(n)}\right\|_{t,l}\right). \tag{10}$$

The first term of the metric measures the average error of ensemble members relative to observations, and the second term provides an unbiased estimate of internal dispersion to correct for finite ensemble size.

The Spread–Skill Ratio evaluates ensemble calibration by comparing the ensemble dispersion to the RMSE of the ensemble mean. The spread is defined as

$$\mathrm{Spread}_l = \sqrt{\frac{1}{TIJ}\sum_{t}^{T}\sum_{i}^{I}\sum_{j}^{J}\mathrm{var}_m\left(f_{t,l,i,j,m}\right)}, \tag{11}$$

with the variance of ensemble members given by

$$\mathrm{var}_m\left(f_{t,l,i,j,m}\right) = \frac{1}{M-1}\sum_{m=1}^{M}\left(f_{t,l,i,j,m} - \bar{f}_{t,l,i,j}\right)^2. \tag{12}$$

The spread–skill ratio is calculated as follows:

$$R_l = \frac{\mathrm{Spread}_l}{\mathrm{RMSE}_l(\bar{f})}. \tag{13}$$

https://doi.org/10.3390/electronics15081583

A value of $R_l \approx 1$ indicates good calibration, where ensemble dispersion reflects the actual prediction error. Values greater than one indicate overestimated uncertainty, while values below one indicate insufficient ensemble variability. Unlike CRPS, which balances error and dispersion, the spread–skill ratio directly measures the calibration of the ensemble [40,41].

In this work, a bias-corrected RMSE is used to reduce systematic errors when calculating the spread–skill ratio, providing a more reliable estimate of ensemble calibration.

4.2. Implementation

The preprocessing pipeline consists of organizing the raw atmospheric, oceanographic, and static data into structured samples based on predefined date ranges for training, validation, and testing. Each sample includes two initial ocean states and the corresponding forcing variables. Before training, all input variables are normalized using precomputed mean and standard deviation values derived from the training set. Static features such as bathymetry and spatial coordinates are also incorporated as additional inputs.

Noise is introduced during the data loading stage by adding perturbations to the initial ocean states only, while forcing variables remain unchanged. Each ensemble member corresponds to a different random realization of the noise, ensuring variability in the predictions.

The code used in this study is publicly available at: https://github.com/Alejeglez/ensemble-gnn-sst, accessed on 5 April 2026.

The Python (version 3.10.16) implementation adds Gaussian noise to each grid point using `numpy.random.normal()` (NumPy version 1.26.4), with configurable mean and standard deviation. Perlin noise is generated using the Python library `perlin-numpy` available at: https://github.com/pvigier/perlin-numpy, accessed on 5 April 2026.

4.3. Training Procedure

We employed a training period spanning 21 years of data with the dataset divided as follows: training data from 2003 to 2019 (17 years), validation data from 2020 to 2021 (2 years), and test data from 2022 to 2023 (2 years). This corresponds to approximately 81% of the data for training and 9.5% for validation and testing.

The model was trained using the AdamW optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay $\lambda = 0.1$. The initial learning rate was set to 1×10^{-5}, with five warm-up epochs and a cosine decay schedule to progressively adjust the learning rate.

Network hyperparameters were selected as follows: hidden layers of 128 neurons, resulting in 128-dimensional vector representations within the GNNs and associated MLPs; four processing layers in the model processor; and a hierarchical mesh with three uniform levels at resolutions of 81, 27, and 9, respectively. Training was conducted for 150 epochs without autoregressive steps due to computational constraints.

The training environment consisted of a desktop workstation with an Intel Core i9-9900K processor (3.60 GHz, 8 cores, 16 threads), 32 GB of RAM, and an NVIDIA GeForce RTX 3060 GPU with 12 GB of VRAM, which enabled efficient handling of large datasets and accelerated computation.

Before training, static features, such as bathymetry, coordinates, and normalization statistics, were computed, and differences between consecutive states were normalized to facilitate residual learning. The structure of the hierarchical mesh and bipartite connections were generated and stored for efficient access.

During training, the model learns to predict one-step-ahead oceanographic states. For evaluation, ensemble predictions are generated by introducing diversity in the initial states through noise perturbations (Gaussian, Perlin, and Perlin fractal). The ensemble predictions

https://doi.org/10.3390/electronics15081583

are then aggregated via daily averages, enabling both probabilistic and deterministic assessment of model performance.

4.4. Test Configurations

A preliminary parameter calibration phase was conducted before the ensemble experiments to define suitable noise configurations and to ensure a structured and reproducible exploration of the parameter space. Rather than performing a dense grid search, we adopt a controlled sensitivity analysis strategy aimed at sampling representative regions of each noise family.

In this phase, candidate parameter values for each noise type were evaluated based on their impact on forecast accuracy. Specifically, a one-factor-at-a-time approach was employed, in which each parameter was varied while the remaining ones were kept fixed, allowing us to isolate its individual effect. Each configuration was evaluated by computing the RMSE at each forecast lead time for 29 randomly selected initialization dates, drawn using a fixed random seed to ensure reproducibility, and then averaging across initialization dates to obtain a representative RMSE curve as a function of lead time.

This procedure defines a parameter screening stage, in which an initial set of candidate values is explored and a reduced number of representative configurations is selected for the subsequent ensemble experiments. The selection is guided by RMSE performance and aims to retain configurations that capture distinct behavioral regimes while avoiding redundancy.

The goal of this study was not to perform an exhaustive exploration of the full parameter space, but to compare qualitatively different types of perturbations under practical computational constraints. Therefore, the selected configurations should be understood as representative cases designed to illustrate differences in noise structure and behavior.

In addition, the parameter values were chosen such that the overall noise intensity remained roughly comparable across Gaussian and Perlin-based perturbations. This allows the comparison to focus on structural differences rather than differences in magnitude. For the Gaussian noise model, the standard deviation of the noise distribution was the only parameter considered for exploration, as it directly controls the amplitude of the perturbations.

The standard deviations used (σ = 0.01, 0.03, 0.05, 0.1, 0.2, 0.5) correspond to approximately 0.5%, 1.4%, 2.3%, 4.5%, 9.1%, and 22.7% of the observed SST variability in the study region (standard deviation $\approx$ 2.2 K). These values were selected as part of an initial systematic exploration of the Gaussian noise amplitude, spanning a wide range of perturbation regimes from weak to strongly forced conditions.

As shown in Figure 8, the best RMSE performance is consistently obtained for the lowest noise amplitudes. A gradual degradation of forecast skill is observed as the standard deviation increases, with a noticeable change in slope around $\sigma = 0.1$, beyond which performance deteriorates more rapidly, and a stronger increase in error is observed for $\sigma \geq 0.2$.

Based on this behavior, the selected configurations (σ = 0.01, 0.05, 0.1) are chosen to represent low to intermediate noise amplitudes within the explored range. In particular, $\sigma = 0.01$ and $\sigma = 0.05$ correspond to weak or near-deterministic perturbations, while $\sigma = 0.1$ lies at the upper end of the low-noise regime. The selected Gaussian configurations are summarized in Table 3.

For the Perlin noise model, two parameters are considered: the spatial scale of the noise and the tileability constraint. The spatial scale is controlled by the number of grid subdivisions used to generate the noise field, which determines the size of the resulting spatial structures. The tileability parameter is set to enforce spatial continuity along the

https://doi.org/10.3390/electronics15081583

latitude dimension, ensuring coherent perturbations across domain boundaries. Spatial coherence is enforced by construction through the tileability constraint.

The resolutions of the Perlin noise were chosen based on the spatial autocorrelation structure and characteristic spatial scales of the SST field in the study region. Analysis of SST anomalies derived from the CMEMS product at 0.05° × 0.05° resolution indicates a dominant spatial scale of approximately 15 pixels (≈83 km), as shown in Figure 9.

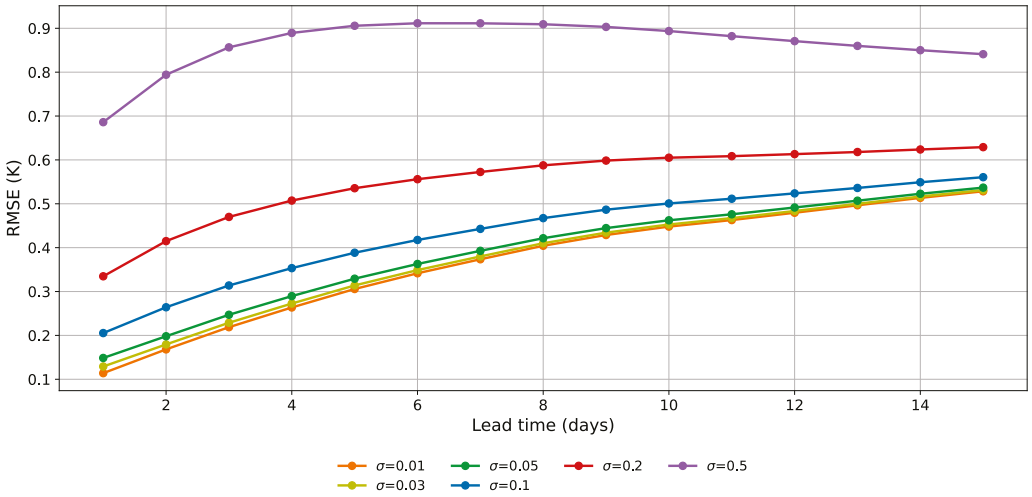


Figure 8. Validation RMSE as a function of forecast lead time for different Gaussian noise standard deviations (σ). Each curve corresponds to a different noise amplitude.

Table 3. Gaussian noise configurations used during the tests.

Noise Type	Mean (μ)	Standard Deviation (σ)
Gaussian ($\mu = 0, \sigma = 0.1$)	0.0	0.1
Gaussian ($\mu = 0, \sigma = 0.05$)	0.0	0.05
Gaussian ($\mu = 0, \sigma = 0.01$)	0.0	0.01

A configuration with 15×15 resolution on the 300 × 300 pixel domain generates perturbations of roughly 20 pixels (≈111 km), comparable in magnitude to the observed SST structures. This ensures that the noise captures physically meaningful spatial patterns. The classic Perlin noise configuration with 12 × 12 resolution produces slightly larger and coarser patterns, allowing evaluation of the effect of pattern size and density on forecast performance. A 3 × 3 base Perlin noise was also included to test very coarse, low-detail patterns.

Based on this physical constraint, the explored configurations are defined by varying the number of spatial subdivisions (3, 5, 6, 12, 15, and 25), which correspond to progressively finer spatial scales. This range spans scales both larger and smaller than the dominant SST variability (15 pixels), enabling a structured exploration of how the noise field interacts with physically relevant spatial scales.

The tileability parameter was also evaluated under a fixed-resolution setting (2 × 6 × 6) to assess its potential influence. However, no significant differences were observed in forecast performance between tileable and non-tileable configurations, suggesting a limited impact within the considered experimental setup. As a result, tileability is not further considered in the remainder of the study.

The RMSE analysis, as shown in Figure 10, indicates that the coarsest configuration (3 subdivisions) yields slightly better performance at early forecast lead times. However, no monotonic relationship is observed between spatial scale and forecast error. In particular,

https://doi.org/10.3390/electronics15081583

the finest-scale configuration does not consistently exhibit the worst performance, and intermediate and fine-scale configurations produce comparable RMSE values. As lead time increases, differences across configurations further diminish, indicating a weak sensitivity of forecast skill to the spatial scale of the perturbations.

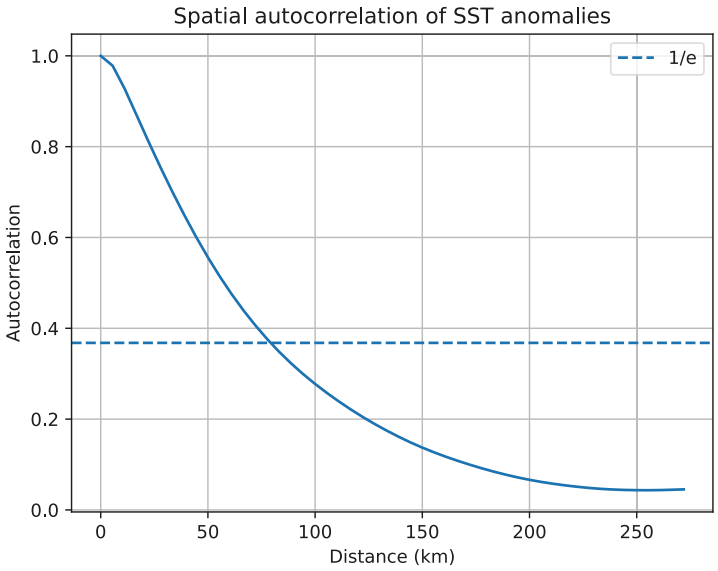


Figure 9. Spatial autocorrelation of SST deviations. The dashed line marks the correlation length, where the autocorrelation drops to $1/e$, indicating the scale of SST structures in the region.

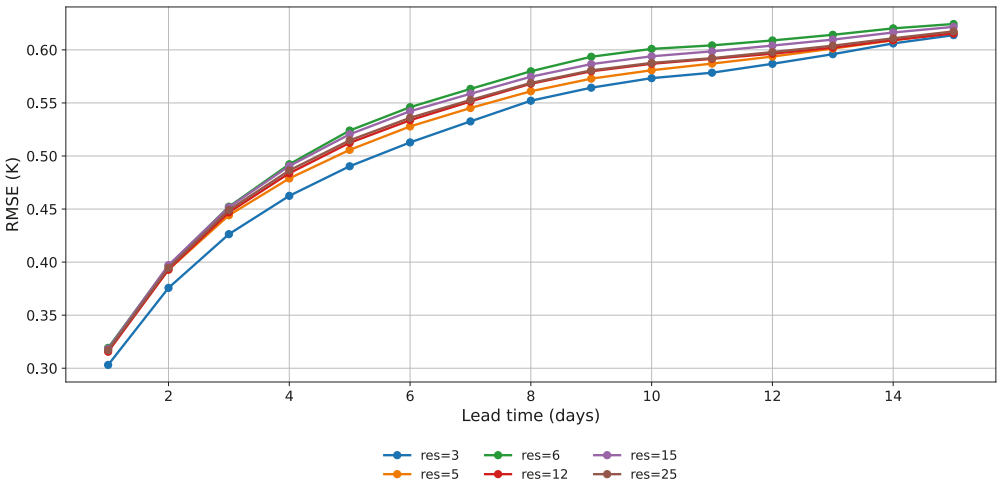


Figure 10. Validation RMSE as a function of forecast lead time for different Perlin noise configurations, defined by the noise resolution. Each curve corresponds to a different spatial scale of the noise field.

Based on this behavior, two representative configurations (3×3 and 12×12 resolutions) are selected for the subsequent evaluation stage. These configurations are chosen to represent contrasting levels of spatial detail: a coarse-scale regime characterized by large spatial structures, and an intermediate-scale regime with finer variability. In addition, they differ from the resolution configurations selected for the fractal Perlin noise model, enabling a clearer comparison between noise types. The selected Perlin configurations are summarized in Table 4, where we use the following systematic naming scheme:

- Classic Perlin noise uses `P_res_XxYxZ`, where `XxYxZ` indicates the spatial resolution along each dimension.

- Fractal Perlin noise uses `PF_res_XxY`, where `XxY` indicates the 2D spatial resolution. Optional modifications are added as suffixes: `(α=value)` indicates a modified scale factor applied to the noise.

Table 4. Classic Perlin noise configurations used during the tests.

Noise Type	Resolution	Repeatable
`P_res_2x3x3`	(2, 3, 3)	(T, F, F)
`P_res_2x12x12`	(2, 12, 12)	(T, F, F)

For the fractal Perlin noise configurations, three parameters are explored: spatial resolution, tileability, and noise scale (α). As in the previous configurations, the tileability constraint does not lead to noticeable differences in the spatial structure of the perturbations under the considered experimental setup and is therefore not further explored in detail.

The baseline configuration is derived from the setup proposed in Pangu-Weather [15], which uses a fractal Perlin noise with 15×15 spatial resolution, three octaves, persistence of 0.5, lacunarity of 2, and a noise scale of 0.2. In this study, octaves, persistence, and lacunarity are kept fixed to the recommended values from Pangu-Weather to preserve the multi-scale structure of the noise and isolate the effect of the selected control parameters.

Spatial resolutions are chosen to ensure compatibility with the iterative octave construction, where each successive octave modifies the effective spatial scale according to the lacunarity factor. Given the 300×300 domain, the studied configurations (5×5, 15×15 and 25×25) are designed to be compatible with the domain discretization and the previous spatial analysis, while providing a comparable setting to the base Perlin configurations.

Unlike standard Perlin noise, which can be formulated in a fully spatio-temporal (3D) framework, the implementation used here is purely two-dimensional in space. Due to the octave-based construction, the noise field is generated independently for each temporal state, and therefore, no explicit coherence is enforced between noise realizations at consecutive time steps. However, as shown in Figure 11, the RMSE analysis does not reveal a consistent sensitivity to resolution within this range, indicating that performance differences are not strongly driven by spatial resolution, as was also observed for the base Perlin configuration. Accordingly, the final analysis focuses on two representative resolutions: 5×5 and 15×15, corresponding respectively to higher and lower levels of spatial subdivision within the tested range.

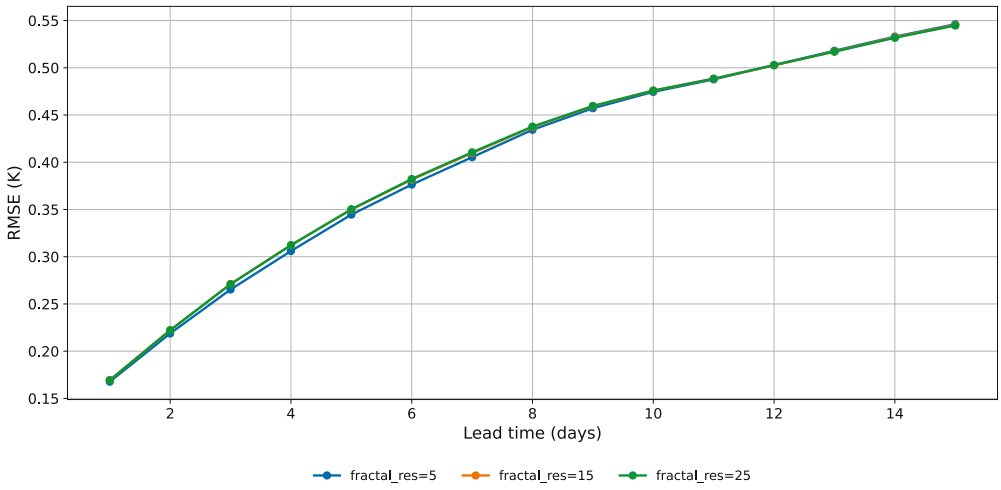


Figure 11. Validation RMSE as a function of forecast lead time for different Perlin fractal noise configurations, defined by the noise resolution. Each curve corresponds to a different spatial scale of the noise field.

The noise scale parameter is analyzed because it directly controls how strong the random perturbations are. Three values are tested: 0.05, 0.2, and 0.4. The smallest value (0.05) represents a weak-perturbation case that is close to deterministic behavior, while 0.2 corresponds to the baseline used in previous work (Pangu-Weather), and 0.4 represents a strongly perturbed case where random effects become dominant. As shown in Figure 12, the RMSE clearly depends on this parameter. Lower noise values consistently lead to better forecast accuracy. Because of this, all three values are kept for the final evaluation to cover the full range of model behavior under different noise strengths. The selected fractal Perlin configurations are summarized in Table 5.

All experiments were conducted using an ensemble-based approach. Unless otherwise specified, the default ensemble size was 5 members, each corresponding to a different random realization of the noise. Preliminary tests were first performed using 29 randomly selected initialization dates from the test dataset, matching those used in the noise analysis stage. This stage was used to assess the relative performance of different noise configurations under multiple initial conditions and to discard clearly underperforming candidates, including probabilistic metrics.

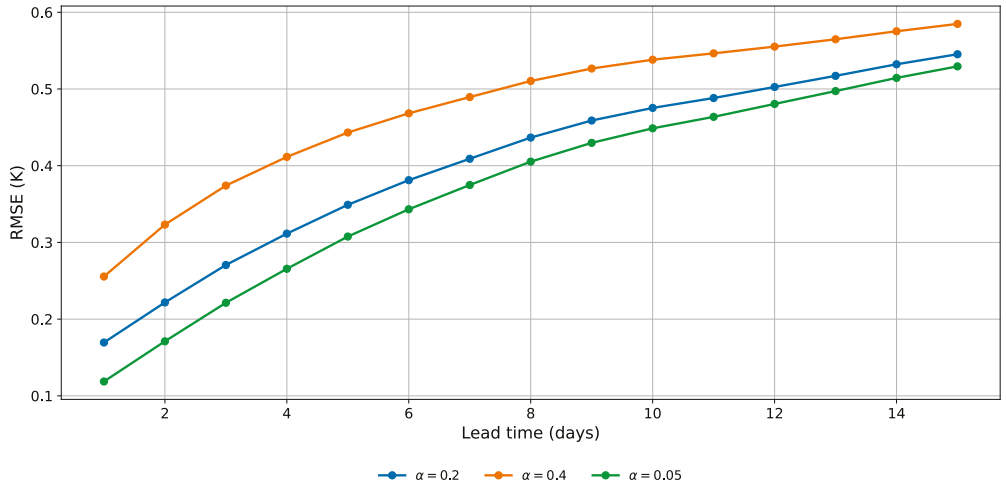


Figure 12. Validation RMSE as a function of forecast lead time for different Perlin fractal noise configurations, defined by the noise scale. Each curve corresponds to a different noise intensity.

Table 5. Fractal Perlin noise configurations used during the tests. The IDs correspond to: (A) `PF_res_15x15`; (B) `PF_res_5x5`; (C) `PF_res_15x15` ($\alpha = 0.05$); and (D) `PF_res_15x15` ($\alpha = 0.4$).

ID	Resolution	Repeatable	Persistence	Octaves	Scale	Lacunarity
A	(15, 15)	(F, T)	0.5	3	0.2	2.0
B	(5, 5)	(F, T)	0.5	3	0.2	2.0
C	(15, 15)	(F, T)	0.5	3	0.05	2.0
D	(15, 15)	(F, T)	0.5	3	0.4	2.0

To characterize variability, results are reported using empirical 95% confidence intervals computed over evaluation metrics across initialization dates and ensemble members, providing an estimate of performance dispersion under stochastic initialization. No formal statistical significance testing is performed. Therefore, comparisons between configurations are interpreted in terms of relative performance trends and observed variability rather than strict statistical significance.

The configurations selected from this preliminary analysis were then evaluated in a full test setting spanning two years, starting on 1 January 2022. This final evaluation uses continuous forecasts initialized throughout the entire test period, ensuring that results

reflect performance across a wide range of meteorological conditions rather than a limited set of initial states.

5. Results

In this section, we compare ensemble prediction with the deterministic forecast. Then, we analyze the differences observed in preliminary experiments using Gaussian and Perlin noise, emphasizing how specific noise parameters influence performance when initial conditions are perturbed.

Table 6 reports the RMSE averaged over 1, 5, and 15 days, relative to the deterministic forecast without noise. None of the ensemble configurations consistently outperforms the deterministic model across all horizons, which is expected given that the model was trained under deterministic conditions.

Table 6. RMSE absolute values and relative increase (%) over different lead times for each noise configuration, with respect to the deterministic model without noise.

Configuration	1 Day	5 Days	15 Days
Deterministic (ref)	0.109 (–)	0.308 (–)	0.586 (–)
Gaussian ($\sigma^2 = 0.01$)	0.109 (+0.55%)	0.308 (+0.03%)	0.586 (+0.00%)
Gaussian ($\sigma^2 = 0.1$)	0.140 (+28.96%)	0.325 (+5.36%)	0.588 (+0.43%)
PF_res_15x15 ($\alpha = 0.05$)	0.110 (+1.56%)	0.309 (+0.16%)	0.586 (+0.02%)
P_res_2x12x12	0.178 (+64.01%)	0.357 (+15.90%)	0.595 (+1.62%)
P_res_2x3x3	0.175 (+61.17%)	0.354 (+14.83%)	0.594 (+1.47%)

RMSE increases with forecast horizon due to the autoregressive formulation. Ensemble predictions also exhibit higher RMSE at short lead times because the input observations are explicitly perturbed. The largest initial degradations are observed for `P_res_2x12x12`, `P_res_2x3x3`, and Gaussian ($\mu = 0$, $\sigma = 0.1$), reflecting their higher input variability. However, this effect decreases with lead time, and performance converges toward stable values by the final forecast day.

Overall, these results indicate that introducing diversity in the initial conditions helps compensate for forecast errors and promotes exploration of alternative future states, which becomes increasingly important as uncertainty grows in autoregressive predictions. The primary objective of this work was not to improve deterministic accuracy, but to evaluate how different ensemble strategies and input perturbations impact uncertainty characterization.

Figure 13 shows the bias on prediction day 1 for different members of the ensemble and their mean value. The members show greater variability, while the mean of the set is more accurate.

5.1. Comparison of Gaussian Noise Configurations

Table 7 summarizes the mean CRPS for the different Gaussian noise configurations across lead-time ranges.

Figure 14 shows the CRPS (10) and its components, CRPS skill (7) and CRPS spread (8), for the selected Gaussian noise configurations. Values closer to zero indicate better performance of the set, as they reflect a balance between accuracy (low error of the predicted distribution) and adequate dispersion among the predictions of the set, which favors prediction sets that explore more alternatives with controlled error.

In this configuration, the CRPS is higher at the beginning of the forecast horizon when the first two entries correspond to noisy initial states. The figure shows that the CRPS skill curve reflects greater disagreement with actual observations in the early steps. Although the contribution of the CRPS spread helps to mitigate the error, it is not sufficient to achieve

the values obtained with lower noise levels. In the latter cases, the lower disturbance of the initial conditions results in better performance in the first days of the forecast.

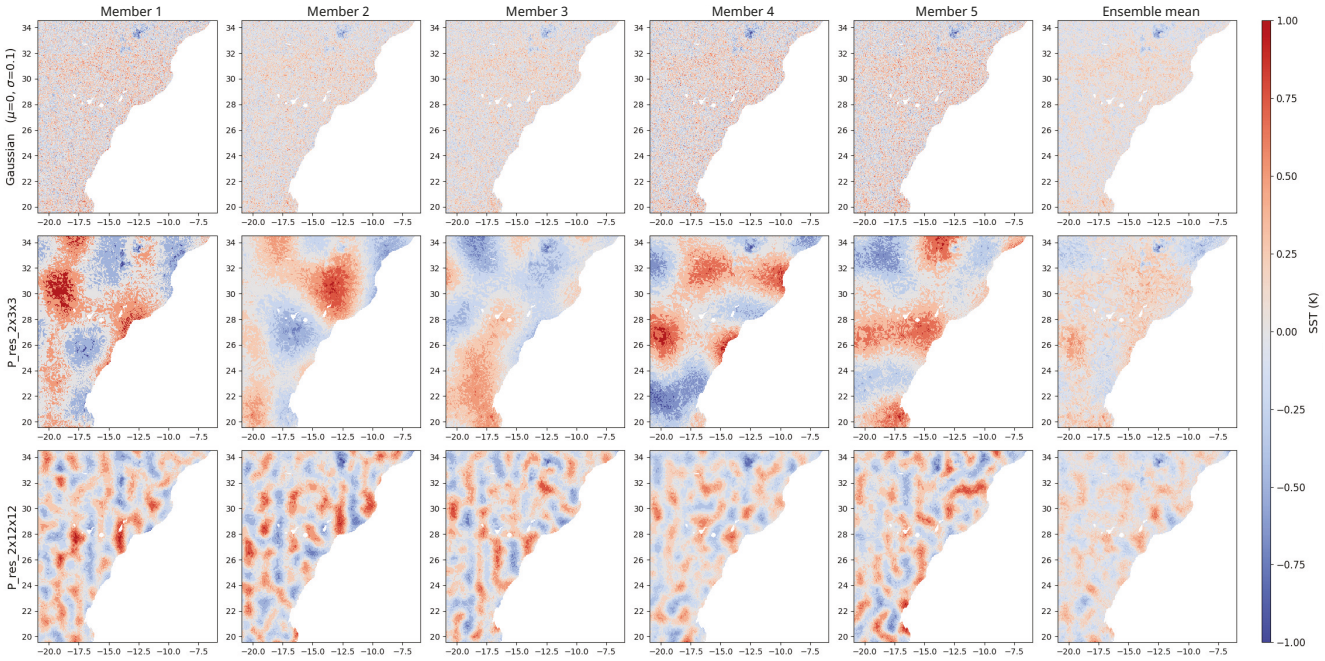


Figure 13. SST bias maps for one-day forecasts starting on 2 January 2022. Each row corresponds to a different input noise configuration, and each column represents a member of the ensemble, with the last column showing the ensemble mean.

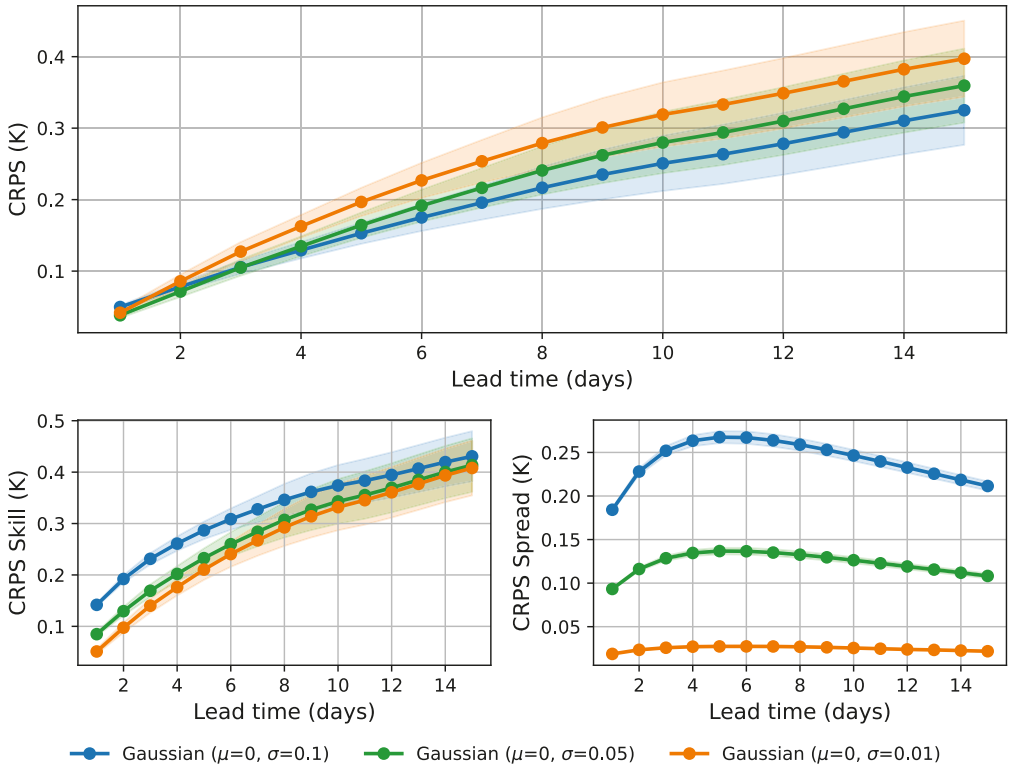


Figure 14. CRPS and its components, averaged over initialization dates for each prediction horizon, with 95% confidence intervals, comparing the different sets of predictions with Gaussian noise.

However, these configurations show a change in trend over the time horizon, reaching their worst performance at the end. This can be understood by analyzing the CRPS skill

https://doi.org/10.3390/electronics15081583

graph, where the differences in error between the configurations are minimal at the end of the time horizon. As their dispersion (CRPS spread) is lower, these errors are not corrected adequately. Therefore, although the dispersion between the members decreases slightly from the sixth day onwards, the model with the highest noise ends up being the one that best represents the predicted distribution compared to the actual observations.

Table 7. Mean CRPS (SST) for Gaussian noise configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
Gaussian ($\mu = 0$, $\sigma = 0.1$)	0.064	0.117	0.164	0.206	0.243	0.271	0.302	0.325
Gaussian ($\mu = 0$, $\sigma = 0.05$)	0.055	0.120	0.178	0.229	0.271	0.302	0.336	0.360
Gaussian ($\mu = 0$, $\sigma = 0.01$)	0.064	0.145	0.212	0.266	0.310	0.341	0.374	0.397

Table 8 summarizes the mean unbiased spread–skill ratio for the different Gaussian noise configurations across lead-time ranges.

Figure 15 depicts the evolution of the unbiased spread–skill ratio (13) and its components: the standard deviation (11) and the unbiased RMSE. The horizontal line at value 1 on the y-axis represents the reference threshold. This would be the case when the dispersion of the predictions of the set with respect to their mean can completely explain the errors of that mean, indicating perfect calibration.

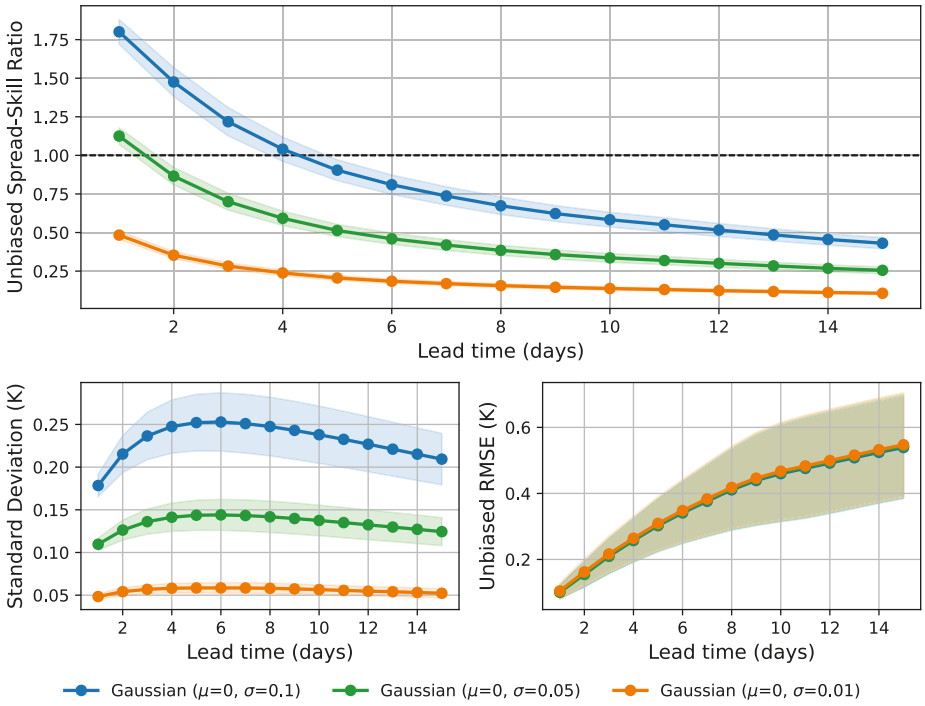


Figure 15. Graph of the unbiased spread–skill ratio and its components, averaged for each prediction horizon, with 95% confidence intervals, comparing different sets of predictions with Gaussian noise introduced.

Table 8. Mean spread–skill ratio (unbiased, SST) for Gaussian noise configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
Gaussian ($\mu = 0$, $\sigma = 0.1$)	1.638	1.129	0.857	0.705	0.603	0.533	0.470	0.430
Gaussian ($\mu = 0$, $\sigma = 0.05$)	0.995	0.646	0.486	0.402	0.346	0.309	0.276	0.255
Gaussian ($\mu = 0$, $\sigma = 0.01$)	0.418	0.260	0.195	0.162	0.141	0.127	0.114	0.106

https://doi.org/10.3390/electronics15081583

In this case, the configuration with standard deviation equal to 0.01 presents a subdispersed behavior throughout the entire prediction horizon, with insufficient dispersion with respect to the error committed by its mean. This can be verified by comparing the graph of standard deviation with respect to RMSE, where it is always a lower value. The other two configurations are overdispersed at the beginning, with greater dispersion with respect to their mean. However, a common pattern is detected in the evolution of the spread–skill ratios, as they begin to stabilize from the sixth day onwards. This is consistent with the fact that the standard deviation of the sets grows until day 6, followed by a progressive decline. Furthermore, at this point it is also detected that the RMSE exceeds the standard deviation, which explains the coefficient of less than one in the case of a standard deviation equal to 0.1. Therefore, although all configurations end up being underdispersed, it can be deduced that with a slightly larger but controlled disturbance, it would be possible to achieve a calibration of the set close to one at the end.

5.2. Comparison of Perlin Noise Configurations

Table 9 summarizes the mean CRPS for the different Perlin noise configurations across lead-time ranges.

The Perlin noise results corresponding to the first days of prediction for CRPS are shown in Figure 16. The configurations with base Perlin noise (P_res_2x3x3 and P_res_2x12x12) show lower initial performance. However, these experiments end up being the best at the end of the time horizon compared to the fractal Perlin configurations, following a similar trend to Gaussian noise ($\mu = 0$, $\sigma^2 = 0.1$). The configurations with Perlin fractal noise start with initial CRPS values close to the ideal, but their performance deteriorates over time.

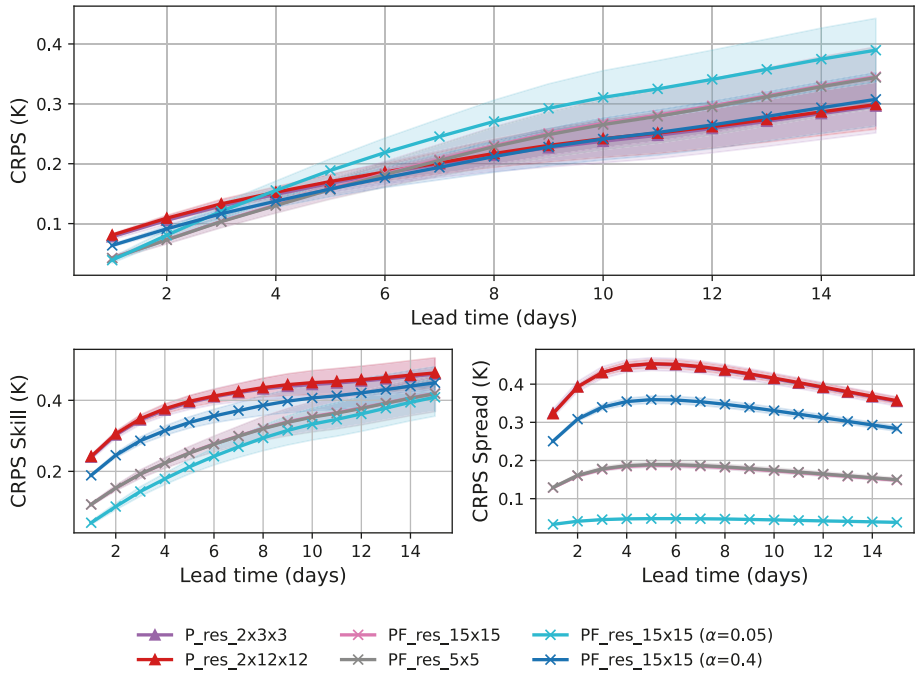


Figure 16. CRPS and its components, averaged for each prediction horizon, with 95% confidence intervals, comparing the predictions with Perlin noise introduced.

One of the differences between the two groups is the complexity of the noise. Perlin fractal noise uses multiple iterations to refine the pattern, each with a higher resolution. However, the influence of these last octaves is minor, so only fine details are added. In contrast, base Perlin noise is simpler, as it generates the noise only once with the input

resolution. Furthermore, fractal Perlin noise configurations incorporate a noise scalability factor, represented by α, which is applied to the output to attenuate its impact.

Table 9. Mean CRPS (SST) for Perlin noise configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
(A)	0.092	0.139	0.176	0.207	0.232	0.253	0.277	0.296
(B)	0.095	0.143	0.178	0.209	0.236	0.257	0.280	0.299
(C)	0.058	0.117	0.171	0.219	0.259	0.289	0.322	0.345
(D)	0.058	0.117	0.169	0.217	0.257	0.287	0.320	0.344
(E)	0.060	0.138	0.204	0.258	0.302	0.333	0.366	0.390
(F)	0.077	0.127	0.167	0.203	0.235	0.258	0.286	0.307

Legend: (A) `P_res_2x3x3`; (B) `P_res_2x12x12`; (C) `PF_res_15x15`; (D) `PF_res_5x5`; (E) `PF_res_15x15` (α = 0.05); (F) `PF_res_15x15` (α = 0.4).

The magnitude of the Perlin fractal noise with a scalability of 0.4 is comparable to the base Perlin noise. However, the dispersion between members is lower in this case, despite using a similar noise intensity. This suggests that fractal Perlin noise is not beneficial due to noise refinement through octaves, which generates complex spatial patterns with less structural coherence, thus affecting uncertainty.

Table 10 summarizes the mean unbiased spread–skill ratio for the different Perlin noise configurations across lead-time ranges.

Regarding the unbiased spread–skill ratio, Figure 17 shows that the basic Perlin configurations stand out once again. Although they exhibit overdispersion over the time horizon, on the 9th day of prediction they reach a value close to one, indicating an almost perfect calibration. Furthermore, it can be seen that in all sets, the standard deviation begins to decrease from the sixth day, as was already the case with Gaussian noise. This decrease, together with a less pronounced increase in unbiased RMSE, causes the spread–skill ratio to stabilize. On the contrary, due to the low intensity of the initial noise, most fractal Perlin configurations end up being underdispersed during most of the prediction period.

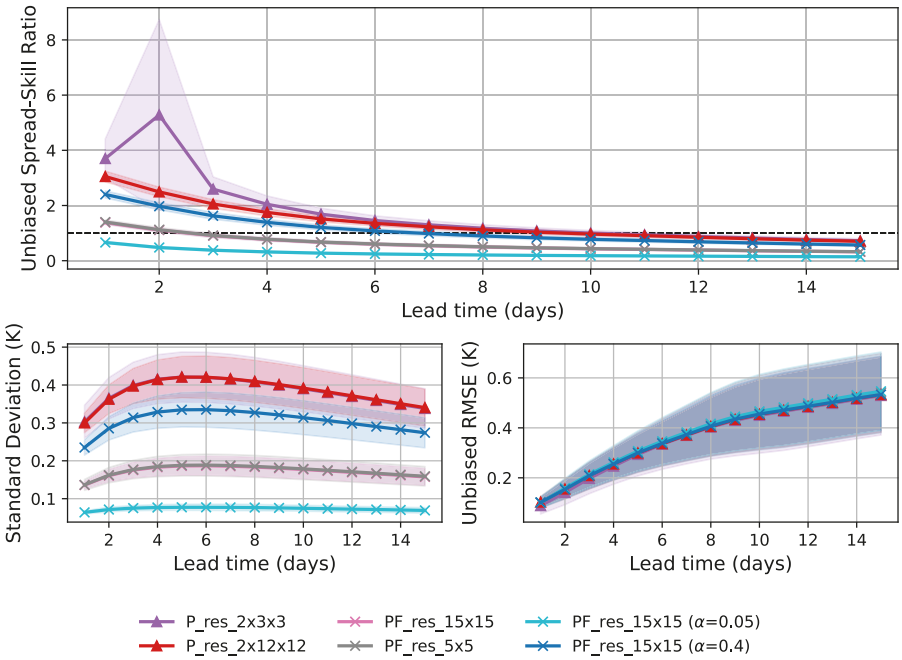


Figure 17. Unbiased spread–skill ratio and its components, averaged for each prediction horizon, with 95% confidence intervals, comparing the different sets of predictions with Perlin noise introduced.

Table 10. Mean spread–skill ratio (unbiased, SST) for Perlin noise configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
(A)	4.492	2.319	1.571	1.238	1.041	0.909	0.795	0.723
(B)	2.772	1.903	1.434	1.174	0.999	0.879	0.770	0.703
(C)	1.243	0.838	0.633	0.521	0.447	0.396	0.351	0.322
(D)	1.261	0.846	0.640	0.528	0.452	0.401	0.355	0.327
(E)	0.569	0.349	0.259	0.215	0.187	0.168	0.152	0.141
(F)	2.187	1.507	1.143	0.940	0.803	0.709	0.623	0.569

Legend: (A) `P_res_2x3x3`; (B) `P_res_2x12x12`; (C) `PF_res_15x15`; (D) `PF_res_5x5`; (E) `PF_res_15x15` (`α = 0.05`); (F) `PF_res_15x15` (`α = 0.4`).

Regarding the RMSE in the unified predictions, Table 11 shows that higher noise levels tend to result in worse performance across the forecast horizon. However, as observed in the previous noise selection phase, differences between Perlin configurations are generally modest, with most approaches converging to similar error values at longer lead times.

Table 11. Mean RMSE (SST) for Perlin noise configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
(A)	0.197	0.289	0.362	0.418	0.457	0.485	0.513	0.534
(B)	0.206	0.299	0.369	0.424	0.466	0.492	0.520	0.540
(C)	0.154	0.250	0.331	0.394	0.443	0.474	0.507	0.530
(D)	0.154	0.250	0.329	0.393	0.441	0.473	0.505	0.528
(E)	0.140	0.241	0.323	0.389	0.438	0.471	0.505	0.528
(F)	0.184	0.276	0.351	0.409	0.453	0.481	0.510	0.532

Legend: (A) `P_res_2x3x3`; (B) `P_res_2x12x12`; (C) `PF_res_15x15`; (D) `PF_res_5x5`; (E) `PF_res_15x15` (`α = 0.05`); (F) `PF_res_15x15` (`α = 0.4`).

5.3. Comparison Between Types of Noise

The most promising noises underwent an additional test in which predictions were made starting on the same days as the previous ones, but with the set size increased to 10. However, in this case, the results did not show significant differences, with minimal changes relative to prior trends. In the final evaluation, the noises with the best performance according to the previous metrics for each group were selected, along with one representative of each type of noise with the worst results for comparison purposes.

Tables 12–14 summarize the mean CRPS, unbiased spread–skill ratio, and RMSE, respectively, for Perlin noise and Gaussian configurations across lead-time ranges.

Figure 18 is consistent with the trends observed in the preliminary experiments, where noise intensity appears important for ensuring sufficient ensemble diversity (e.g., for a Gaussian standard deviation of 0.1). Unlike the preliminary analysis based on single initialization dates, the results shown here are obtained by averaging the metrics over all prediction start dates in the test period (2022–2023), providing a broader assessment. Under this evaluation, overall performance decreases, and the differences between configurations become smaller. Nevertheless, Perlin noise still shows better performance than the Gaussian noise configurations.

Despite the difference in the statistical distribution of noise between the configurations of both groups, the noise intensities are similar in those with comparable performance (e.g., Gaussian noise $\sigma = 0.1$ and Perlin base P_res_12x12). This suggests that the models benefit from initial perturbations that combine adequate noise intensity with some spatial structure.

https://doi.org/10.3390/electronics15081583

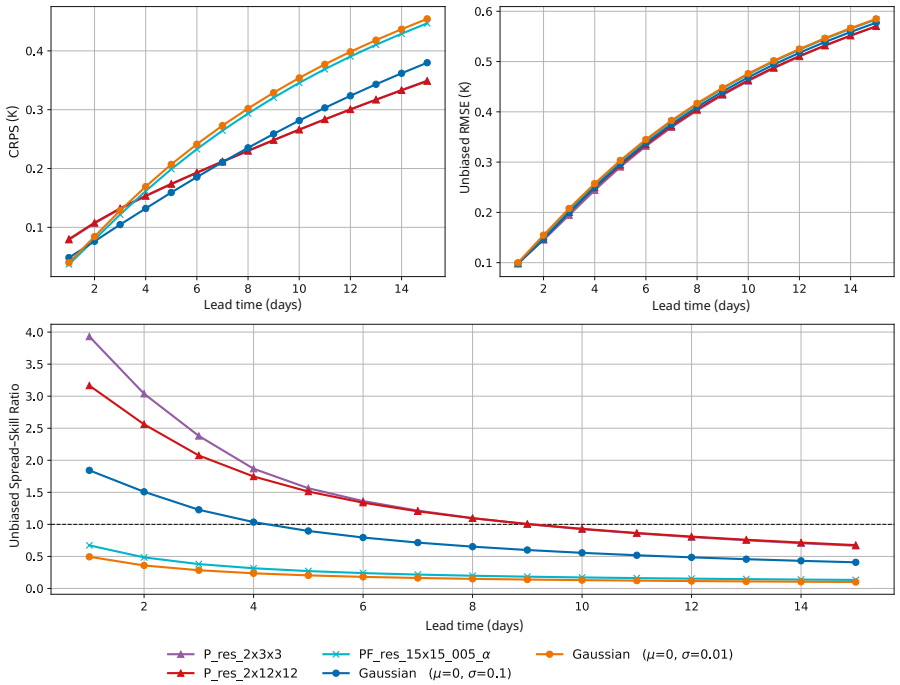


Figure 18. CRPS, unbiased RMSE, and spread–skill ratio averaged by prediction horizon, comparing the different sets of predictions with added noise, evaluated on the complete test data set.

Table 12. Mean CRPS (SST) for Perlin noise and Gaussian configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
A	0.093	0.142	0.183	0.221	0.257	0.292	0.325	0.349
B	0.094	0.143	0.184	0.221	0.257	0.292	0.325	0.349
C	0.058	0.142	0.216	0.279	0.333	0.380	0.420	0.447
D	0.062	0.118	0.172	0.223	0.270	0.313	0.353	0.380
E	0.062	0.149	0.224	0.287	0.342	0.388	0.428	0.454

Legend: (A) `P_res_2x3x3`; (B) `P_res_2x12x12`; (C) `PF_res_15x15` ($\alpha = 0.05$); (D) `Gaussian` ($\mu = 0$, $\sigma^2 = 0.1$); (E) `Gaussian` ($\mu = 0$, $\sigma^2 = 0.01$).

Table 13. Mean Spread–skill ratio (unbiased, SST) for Perlin noise and Gaussian configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
A	3.484	2.124	1.464	1.156	0.964	0.831	0.729	0.666
B	2.863	1.911	1.425	1.149	0.968	0.838	0.738	0.676
C	0.579	0.348	0.256	0.208	0.178	0.158	0.142	0.133
D	1.676	1.132	0.846	0.684	0.578	0.502	0.444	0.408
E	0.427	0.261	0.193	0.157	0.135	0.119	0.108	0.101

Legend: (A) `P_res_2x3x3`; (B) `P_res_2x12x12`; (C) `PF_res_15x15` ($\alpha = 0.05$); (D) `Gaussian` ($\mu = 0$, $\sigma^2 = 0.1$); (E) `Gaussian` ($\mu = 0$, $\sigma^2 = 0.01$).

Table 14. Mean RMSE (unbiased, SST) for Perlin noise and Gaussian configurations across lead-time ranges.

Noise Type	1–2d	3–4d	5–6d	7–8d	9–10d	11–12d	13–14d	15–16d
A	0.122	0.219	0.311	0.386	0.447	0.498	0.541	0.570
B	0.123	0.225	0.315	0.388	0.449	0.500	0.542	0.570
C	0.124	0.230	0.322	0.398	0.460	0.512	0.555	0.584
D	0.122	0.225	0.317	0.393	0.455	0.506	0.549	0.577
E	0.127	0.233	0.324	0.400	0.462	0.513	0.556	0.585

Legend: (A) `P_res_2x3x3`; (B) `P_res_2x12x12`; (C) `PF_res_15x15` ($\alpha = 0.05$); (D) `Gaussian` ($\mu = 0$, $\sigma^2 = 0.1$); (E) `Gaussian` ($\mu = 0$, $\sigma^2 = 0.01$).

https://doi.org/10.3390/electronics15081583

6. Discussion

While deterministic error remains comparable to the single-model forecast, the predictions reveal important patterns in ensemble performance and uncertainty, providing a basis for further investigation.

The analysis of the initial perturbations shows differences depending on their structure and intensity. Perturbations generated with moderate noise and structured spatial patterns (such as Perlin noise) appear more effective than those generated with unstructured noise (e.g., Gaussian noise), which does not incorporate explicit spatial structure. Furthermore, iterating over spatial noise patterns by introducing finer-scale details in successive steps (as in Perlin fractal noise) did not lead to noticeable improvements.

A limitation of this study is that, although the noise configurations are explored through a structured sensitivity analysis, the parameter space is not exhaustively sampled in a global sense. Instead, we focus on a set of representative configurations selected from predefined parameter ranges in order to enable controlled comparisons of different perturbation structures under computational constraints. While this ensures a systematic and interpretable analysis, it may still omit configurations that could further improve performance or reveal additional behaviors.

In addition to these findings, it is important to consider the characteristics of the base model when interpreting the results. The model is relatively simple, both in terms of the number of variables and the training procedure, and it was trained deterministically. Despite this, some predictions obtained by perturbing the initial state perform similarly to the original, unperturbed model in this specific case. This suggests that adding small changes at inference time can introduce some forecast variability without changing the training process. However, more work is needed to understand whether this result would hold for more complex models or datasets. Although ensemble predictions tend to perform worse at early stages, differences between individual predictions seem to balance out over time, leading to an average result similar to the deterministic model.

The current model is limited by the use of the SST and surface winds, which restricts its ability to capture complex oceanographic processes, such as coastal upwelling, mesoscale circulation, or vertical mixing. Incorporating additional physical variables in future work could improve the dynamical realism of forecasts and allow ensemble perturbations to better represent variability from these processes.

Future work could explore incorporating CMEMS error estimates to scale ensemble perturbations or weight training samples. Doing so would allow the ensemble to more accurately reflect observational uncertainty, particularly in regions with strong variability such as the Canary Islands upwelling region.

It is important to note that the model was trained using a single-step objective (T = 1) and extended to 15-day forecasts via autoregressive rollout during testing. This discrepancy between the training and inference horizons is a known limitation of one-step training and can contribute to error accumulation at longer lead times. The primary goal of this work is not to optimize the forecasting model itself, but to evaluate how ensemble perturbation strategies influence prediction diversity on a fixed baseline. Incorporating multi-step rollouts during training could improve long-term stability and represents a potential direction for future work.

This study identified several issues for further research beyond the exclusive perturbation of initial oceanographic data during inference. One possible direction is to train the model with autoregressive steps, which could improve the accuracy of long-term predictions. In this work, noise was added exclusively to the initial oceanographic states of the model. Considering that only one variable was used, this diversity may not be sufficient to capture complex phenomena such as coastal upwelling, which probably requires a greater

https://doi.org/10.3390/electronics15081583

number of variables to explain adequately. This scenario would also allow us to validate the conclusions drawn in this work regarding the types of noise that are most beneficial for creating larger-scale prediction sets.

Similarly, other types of ensemble learning techniques could be used for comparison, including the use of lagged predictions, the perturbation of initial forcing states, or the modification of the model parameters. As one of the advantages of ensemble learning techniques is their great flexibility, future work could focus on introducing diversity in the training phase rather than inference, provided the necessary computational resources are available. Although no significant improvements were obtained in this work by increasing the number of ensemble members, this is likely due to the small number of predictions used, conditioned by computational limitations. This deficiency was evident in the analysis of the unbiased RMSE, where in several scenarios the results changed dramatically.

Therefore, this work aims to serve as an initial reference in the study of the types of noise applied during the inference phase as a mechanism for introducing diversity into prediction sets. However, as discussed throughout this section, the possibilities for configuring and incorporating diversity in this type of technique are extensive. As a result, future lines of research could focus on improving the results obtained in this work using the same techniques or exploring any of the other options mentioned.

Author Contributions: Conceptualization, G.A.C.-L. and J.S.; methodology, J.S.; software, A.J.G.-S.; validation, A.J.G.-S., G.A.C.-L. and J.S.; formal analysis, G.A.C.-L. and J.S.; investigation, A.J.G.-S., G.A.C.-L. and J.S.; resources, J.S.; data curation, A.J.G.-S. and G.A.C.-L.; writing—original draft preparation, A.J.G.-S., G.A.C.-L. and J.S.; writing—review and editing, A.J.G.-S., G.A.C.-L. and J.S.; supervision, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is developed with the collaboration of the Biodiversity Foundation of the Ministry for Ecological Transition and the Demographic Challenge, through the Pleamar Programme, and is co-financed by the European Union through the European Maritime, Fisheries and Aquaculture Fund through the project *Integrated Remote System with Advanced Tools for Monitoring, Detection and Prediction of Risks from Potentially Harmful Marine Events of Natural or Anthropogenic Origin in Offshore Aquaculture Areas* (SIRENA 2).

Data Availability Statement: The data presented in this study are openly available in: Copernicus Climate Change Service, Climate Data Store, (2023) at 10.24381/cds.adbb2d47, entitled "ERA5 hourly data on single levels from 1940 to present"; NOAA National Centers for Environmental Information, (2022) at 10.25921/fd45-gt74, entitled "ETOPO 2022 30 arc-second bedrock elevation geotiff"; and Copernicus Marine Service, (2023) at 10.48670/moi-00153, entitled "European North West Shelf/Iberia Biscay Irish Seas—High Resolution L4 Sea Surface Temperature". The source code developed for this work is publicly available at https://github.com/Alejeglez/ensemble-gnn-sst, accessed on 5 April 2026, under an MIT license.

Acknowledgments: During the preparation of this manuscript/study, the author(s) used ChatGPT (OpenAI, GPT-4o; accessed on 5 April 2026 via https://chat.openai.com), Gemini (Google, Flash model; accessed on 5 April 2026 via https://gemini.google.com), and Perplexity AI (GPT-4o-based system; accessed on 5 April 2026 via https://www.perplexity.ai) for the purposes of generating text, analysis, and interpretation of data. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CRPS	Continuous Ranked Probability Score
GNN	Graph Neural Network

RMSE Root Mean Square Error
SST Sea Surface Temperature

Appendix A. Comparison Between Single-Step and Autoregressive Models

Both models were trained using the same architecture, hyperparameters, and training procedure, as described in Section 4.3, ensuring a consistent basis for comparison. During validation, predictions are performed over up to four autoregressive steps for both models, which may introduce error accumulation over longer horizons.

On the validation dataset (2020–2021), the autoregressive model converges after approximately 90 epochs with a mean RMSE of 2.33 K, while the single-step model converges after approximately 140 epochs with a mean RMSE of 2.37 K, corresponding to a difference of 0.04 K.

In terms of computational cost, inference on the test dataset (2022–2023) requires approximately 12 h for the autoregressive model, compared to 2.5 h for the single-step model. These results indicate only minor performance differences between the two models, while the computational cost is substantially higher for the autoregressive model.

References

1. Novellino, A.; Arnaud, A.; Schiller, A.; Wan, L. End User Applications for Ocean Forecasting: Present status description. *State Planet Discuss.* **2024**, *2024*, 1–6.
2. Veitch, J.; Alvarez-Fanjul, E.; Capet, A.; Ciliberti, S.; Cirano, M.; Clementi, E.; Davidson, F.; el Serafy, G.; Franz, G.; Hogan, P.; et al. A description of ocean forecasting applications around the globe. *State Planet* **2025**, *5-opsr*, 6. [CrossRef]
3. Fox-Kemper, B.; Adcroft, A.; Böning, C.W.; Chassignet, E.P.; Curchitser, E.; Danabasoglu, G.; Eden, C.; England, M.H.; Gerdes, R.; Greatbatch, R.J.; et al. Challenges and Prospects in Ocean Circulation Models. *Front. Mar. Sci.* **2019**, *6*, 65. [CrossRef]
4. Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N.; Prabhat, F. Deep learning and process understanding for data-driven Earth system science. *Nature* **2019**, *566*, 195–204. [CrossRef]
5. Battaglia, P.; Hamrick, J.B.C.; Bapst, V.; Sanchez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. *Relational Inductive Biases, Deep Learning, and Graph Networks*; Technical Report; Cornell University: New York, NY, USA, 2018.
6. Lam, R.; Sanchez-Gonzalez, A.; Willson, M.; Wirnsberger, P.; Fortunato, M.; Alet, F.; Ravuri, S.; Ewalds, T.; Eaton-Rosen, Z.; Hu, W.; et al. Learning skillful medium-range global weather forecasting. *Science* **2023**, *382*, 1416–1421. [CrossRef]
7. Oskarsson, J.; Landelius, T.; Lindsten, F. Graph-based Neural Weather Prediction for Limited Area Modeling. In Proceedings of the NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning, New Orleans, LA, USA, 16 December 2023.
8. Holmberg, D.; Clementi, E.; Epicoco, I.; Roos, T. Accurate Mediterranean Sea forecasting via graph-based deep learning. *Sci. Rep.* **2025**, *15*, 45051. [CrossRef]
9. Cuervo-Londoño, G.A.; Sánchez, J.; Rodríguez-Santana, Á. Deep Learning Weather Models for Subregional Ocean Forecasting: A Case Study on the Canary Current Upwelling System. *Ocean. Model.* 2026, *Preprint*.
10. Gneiting, T.; Katzfuss, M. Probabilistic Forecasting. *Annu. Rev. Stat. Appl.* **2014**, *1*, 125–151. [CrossRef]
11. Zhou, Z.H.; Wu, J.; Tang, W. Ensembling neural networks: Many could be better than all. *Artif. Intell.* **2002**, *137*, 239–263. [CrossRef]
12. Alexe, M.; Lang, S.; Clare, M.; Leutbecher, M.; Roberts, C.; Magnusson, L.; Chantry, M.; Adewoyin, R.; Prieto-Nemesio, A.; Dramsch, J.; et al. *Data-Driven Ensemble Forecasting with the AIFS*; Technical Report 181; ECMWF: London, UK, 2024. [CrossRef]
13. Price, I.; Sanchez-Gonzalez, A.; Alet, F.; Andersson, T.R.; El-Kadi, A.; Masters, D.; Ewalds, T.; Stott, J.; Mohamed, S.; Battaglia, P.; et al. Probabilistic weather forecasting with machine learning. *Nature* **2025**, *637*, 84–90. [CrossRef]
14. Kochkov, D.; Yuval, J.; Langmore, I.; Norgaard, P.; Smith, J.; Mooers, G.; Klöwer, M.; Lottes, J.; Rasp, S.; Düben, P.; et al. Neural general circulation models for weather and climate. *Nature* **2024**, *632*, 1060–1066. [CrossRef]
15. Bi, K.; Xie, L.; Zhang, H.; Chen, X.; Gu, X.; Tian, Q. Accurate medium-range global weather forecasting with 3D neural networks. *Nature* **2023**, *619*, 533–538. [CrossRef]
16. Dietterich, T.G. Ensemble methods in machine learning. In *Proceedings of the International Workshop on Multiple Classifier Systems, Cagliari, Italy, 21–23 June 2000*; Springer: Berlin/Heidelberg, Gremany, 2000; pp. 1–15. [CrossRef]

17. Kuncheva, L.I.; Whitaker, C.J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.* **2003**, *51*, 181–207. [CrossRef]
18. Rasp, S.; Dueben, P.D.; Scher, S.; Weyn, J.A.; Mouatadid, S.; Thuerey, N. WeatherBench: A benchmark data set for data-driven weather forecasting. *J. Adv. Model. Earth Syst.* **2020**, *12*, e2020MS002203. [CrossRef]
19. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
20. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wires Data Min. Knowl. Discov.* **2018**, *8*, e1249. [CrossRef]
21. Yang, Y.; Lv, H.; Chen, N. A survey on ensemble learning under the era of deep learning. *Artif. Intell. Rev.* **2023**, *56*, 5545–5589. [CrossRef]
22. Couairon, G.; Singh, R.; Charantonis, A.; Lessig, C.; Monteleoni, C. Archesweather & Archesweathergen: A deterministic and generative model for efficient ML weather forecasting. *arXiv* **2024**, arXiv:2412.12971
23. Bodnar, C.; Bruinsma, W.P.; Lucic, A.; Stanley, M.; Allen, A.; Brandstetter, J.; Garvan, P.; Riechert, M.; Weyn, J.A.; Dong, H.; et al. A Foundation Model for the Earth System. *Nature* **2025**, *641*, 1180–1187. [CrossRef]
24. Medina, V.; Cuervo-Londoño, G.A.; Sánchez, J. *Leveraging an Atmospheric Foundational Model for Subregional Sea Surface Temperature Forecasting*; Technical Report; Cornell University: New York, NY, USA, 2025. [CrossRef]
25. Perlin, K. An image synthesizer. *ACM Siggraph Comput. Graph.* **1985**, *19*, 287–296. [CrossRef]
26. Sanchez-Lengeling, B.; Reif, E.; Pearce, A.; Wiltschko, A.B. A gentle introduction to graph neural networks. *Distill* **2021**, *6*, e33. [CrossRef]
27. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
28. Keisler, R. Forecasting global weather with graph neural networks. *arXiv* **2022**, arXiv:2202.07575. [CrossRef]
29. Reyes, J.G.; Cuervo-Londoño, G.A.; Sánchez, J. Adaptive Meshes in Graph Neural Networks for Predicting Sea Surface Temperature Through Remote Sensing. In *Proceedings of the Computer Analysis of Images and Patterns*; Castrillón-Santana, M., Travieso-González, C.M., Deniz Suarez, O., Freire-Obregón, D., Hernández-Sosa, D., Lorenzo-Navarro, J., Santana, O.J., Eds.; Springer: Cham, Switzerland, 2026; pp. 361–372.
30. Cuervo-Londoño, G.A.; Reyes, J.G.; Rodríguez-Santana, A.; Sánchez, J. Voronoi-Induced Artifacts from Grid-to-Mesh Coupling and Bathymetry-Aware Meshes in Graph Neural Networks for Sea Surface Temperature Forecasting. *Electronics* **2025**, *14*, 4841. [CrossRef]
31. Wei, W.; Qiao, M.; Jadav, D. GNN-ensemble: Towards random decision graph neural networks. In *Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 18 December 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 956–965.
32. Wong, Z.H.; Yue, L.; Yao, Q. *Ensemble Learning for Graph Neural Networks*; Technical Report; Cornell University: New York, NY, USA, 2023.
33. Oskarsson, J.; Landelius, T.; Deisenroth, M.P.; Lindsten, F. Probabilistic Weather Forecasting with Hierarchical Graph Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2024; Volume 37, pp. 41577–41648. [CrossRef]
34. Cuervo-Londoño, G.A.; Sánchez, J.; Rodríguez-Santana, Á. Forecasting Sea Surface Temperature from Satellite Images with Graph Neural Networks. In *Proceedings of the Computer Analysis of Images and Patterns*; Castrillón-Santana, M., Travieso-González, C.M., Deniz Suarez, O., Freire-Obregón, D., Hernández-Sosa, D., Lorenzo-Navarro, J., Santana, O.J., Eds.; Springer: Cham, Switerland, 2026; pp. 329–339.
35. Zamo, M.; Naveau, P. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Math. Geosci.* **2018**, *50*, 209–234. [CrossRef]
36. Rasp, S.; Hoyer, S.; Merose, A.; Langmore, I.; Battaglia, P.; Russell, T.; Sanchez-Gonzalez, A.; Yang, V.; Carver, R.; Agrawal, S.; et al. WeatherBench 2: A Benchmark for the Next Generation of Data-Driven Global Weather Models. *J. Adv. Model. Earth Syst.* **2024**, *16*, e2023MS004019. [CrossRef]
37. Battaglia, P.; Pascanu, R.; Lai, M.; Rezende, D.J.; Kavukcuoglu, K. Interaction networks for learning about objects, relations and physics. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 5–10 December 2016; NIPS'16, pp. 4509–4517.
38. Chen, L.; Zhong, X.; Zhang, F.; Cheng, Y.; Xu, Y.; Qi, Y.; Li, H. FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *NPJ Clim. Atmos. Sci.* **2023**, *6*, 190. [CrossRef]
39. Pathak, J.; Subramanian, S.; Harrington, P.; Raja, S.; Chattopadhyay, A.; Mardani, M.; Kurth, T.; Hall, D.; Li, Z.; Azizzadenesheli, K.; et al. *FourCastNet: A Global Data-Driven High-Resolution Weather Model Using Adaptive Fourier Neural Operators*; Technical Report; Cornell University: New York, NY, USA, 2022.

40. Fortin, V.; Abaza, M.; Anctil, F.; Turcotte, R. Why should ensemble spread match the RMSE of the ensemble mean? *J. Hydrometeorol.* **2014**, *15*, 1708–1713. [CrossRef]
41. Wilks, D. On the reliability of the rank histogram. *Mon. Weather. Rev.* **2011**, *139*, 311–316. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

AI-Assisted Value Investing: A Human-in-the-Loop Framework for Prompt-Guided Financial Analysis and Decision Support

Andrea Caridi [1], Marco Giovannini [2] and Lorenzo Ricciardi Celsi [1,*]

[1] Dipartimento di Ingegneria e Scienze, Universitas Mercatorum, Piazza Mattei, 10, 00186 Roma, Italy; andreacaridi.ac2@gmail.com
[2] Independent Researcher, 20121 Milan, Italy; giovanninimarco@gmail.com
* Correspondence: lorenzo.ricciardicelsi@unimercatorum.it

Abstract

Value investing remains grounded in intrinsic value estimation, margin-of-safety reasoning, and disciplined fundamental analysis, but its practical execution is increasingly constrained by the scale, heterogeneity, and velocity of modern financial information. Recent advances in artificial intelligence (AI), particularly large language models and automated information-extraction systems, create new opportunities to accelerate financial analysis; however, their outputs remain probabilistic, context-dependent, and potentially error-prone, making governance and verification essential. This article proposes an AI-assisted value investing framework that integrates automated extraction, valuation modeling, explainability, and human-in-the-loop (HITL) supervision into a unified decision-support architecture. The framework is organized into three layers: (i) a data layer for traceable extraction and normalization of structured and unstructured financial information; (ii) a modeling layer for automated key performance indicator (KPI) computation, forecasting support, and discounted cash flow (DCF) valuation; and (iii) an explainability and governance layer for traceability, verification, model-risk control, and analyst oversight. A central contribution of the paper is the operational characterization of prompt literacy as a determinant of analytical reliability, showing that structured, context-aware prompts materially affect extraction correctness, usability, and verification effort. The framework is evaluated through a case study using Rivanna AI on three large U.S. beverage firms—namely, The Coca-Cola Company, PepsiCo, and Keurig Dr Pepper—selected as a controlled, information-rich setting for comparative analysis. The results indicate that the proposed workflow can reduce end-to-end analysis time from approximately 25–40 h in a traditional manual process to approximately 8–12 h in an AI-assisted setting, including citation/source verification, unit and period reconciliation, and review of key valuation assumptions. Rather than eliminating analyst effort, AI shifts it from manual information processing toward verification, adjudication, and interpretation. Overall, the findings position AI not as an autonomous decision-maker, but as a governed reasoning accelerator whose effectiveness depends on structured human guidance, traceability, and disciplined validation. In value investing, a discipline traditionally grounded in labor-intensive fundamental analysis and disciplined intrinsic value estimation, AI introduces the potential to scale analytical coverage and accelerate evidence synthesis. However, AI systems in financial contexts are probabilistic, context-sensitive, and inherently dependent on human interaction, raising critical questions about reliability, governance, and operational integration. This article proposes a structured framework for AI-driven value investing that preserves the foundational principles of intrinsic value, margin of safety, and economic reasoning, while redesigning the analytical workflow through automation, explainability, and human-in-the-loop (HITL) supervision. The proposed architecture integrates three layers: (i) an AI-enabled data

layer for traceable extraction and normalization of structured and unstructured financial information; (ii) a modeling and valuation layer combining automated KPI computation, machine learning forecasting, and discounted cash flow (DCF) valuation; and (iii) an explainability and governance layer ensuring traceability, verification, and model risk control. A central contribution of this work is the operational characterization of prompt literacy, namely the ability to formulate structured, context-aware requests to AI systems, as a critical determinant of system reliability and analytical correctness. Through a focused case study using an AI-assisted analysis platform (Rivanna AI) on three U.S. beverage firms, we provide evidence that structured prompt formulation can improve extraction consistency, reduce verification overhead, and increase workflow efficiency in a human-supervised setting. In this setting, analysis time decreased from a manual range of approximately 25–40 h to 8–12 h with AI assistance and HITL validation, while preserving traceability and decision accountability. The reported hour savings should be interpreted as conservative estimates from the initial deployment phase; additional efficiency gains are expected as operational maturity increases, driven by learning-economy effects. The findings position AI not as an autonomous decision-maker but as a probabilistic reasoning accelerator whose effectiveness depends on structured human guidance, verification discipline, and prompt-driven interaction. These results redefine the role of the financial analyst from manual data processor to reasoning architect, responsible for designing, guiding, and validating AI-assisted analytical workflows.

Keywords: value investing; fundamental analysis; decision support systems; financial AI; NLP; forecasting; explainable AI; human-in-the-loop; model risk; strategic investing

1. Introduction

Value investing, formalized by Graham and Dodd and further developed by subsequent practitioners and scholars, is grounded in a fundamental principle: financial securities should be evaluated based on intrinsic economic value rather than market price alone. This approach requires integrating quantitative financial analysis, qualitative business assessment, and forward-looking valuation models, traditionally through labor-intensive manual workflows involving financial statement analysis, ratio computation, and discounted cash flow (DCF) modeling.

While this methodology remains conceptually robust, its practical execution is increasingly challenged by structural changes in the financial information environment. Modern analysts must process vast and heterogeneous datasets, including regulatory filings, earnings call transcripts, alternative data sources, and macroeconomic signals, under time constraints that limit analytical depth and coverage. These constraints create operational bottlenecks and introduce cognitive risks, including confirmation bias, anchoring, and selective information processing.

Recent advances in artificial intelligence (AI), particularly in natural language processing, automated information extraction, and generative reasoning models, offer the potential to fundamentally reshape financial analysis workflows. AI systems can rapidly extract structured data from unstructured sources, compute financial metrics at scale, synthesize qualitative information, and assist in valuation modeling. However, these systems are not deterministic computational tools. Their outputs are probabilistic and depend on context, prompt formulation, and human interaction.

This creates an important operational challenge. The reliability of AI outputs depends not only on model architecture and training data but also on how clearly and completely the

https://doi.org/10.3390/electronics15061155

analyst formulates the request. In practical terms, prompt quality directly affects extraction accuracy, logical consistency, and analytical usability. For this reason, prompt literacy becomes a relevant operational skill in AI-assisted financial analysis. Poorly specified prompts can produce incomplete or inconsistent outputs, while structured and context-rich prompts improve reliability and efficiency in the considered case study.

This observation motivates a conceptual shift in how AI should be understood within financial analysis. Rather than replacing human analysts, AI systems function as cognitive amplifiers that extend analytical capacity while remaining dependent on structured human guidance and verification. In this augmented intelligence paradigm, the analyst's role evolves from manual data processor to reasoning architect, designing analytical workflows, structuring prompts, validating outputs, and integrating results into coherent investment decisions.

This article proposes a reference framework for AI-driven value investing that integrates automated extraction, valuation modeling, explainability, and human-in-the-loop (HITL) governance into a unified operational architecture. The framework explicitly incorporates prompt literacy as a core operational competence, recognizing that effective interaction with AI systems is a prerequisite for reliable analytical outcomes. This way, we address the *engineering of reliable AI-enabled decision-support systems*, which is a core problem lying at the intersection between modern electronic and computer-based systems, on the one hand, and finance theory, on the other hand. We propose a modular, auditable architecture for integrating large language models into a real-world workflow, emphasizing system-level requirements such as traceability of numerical outputs to primary sources, schema-constrained extraction, cross-model validation, and human-in-the-loop governance gates to manage model risk and operational drift. By framing LLMs as components within a verifiable pipeline (with explicit failure modes, monitoring, and escalation rules), the work contributes practical design patterns for dependable AI deployment in software-intensive electronic systems where correctness, accountability, and reproducibility are critical.

We provide evidence of this framework through a case study using an AI-assisted financial analysis platform (Rivanna AI), applied to comparative valuation of major US beverage companies. The case study evaluates extraction correctness, valuation consistency, and productivity improvements relative to traditional manual workflows while analyzing the operational role of structured prompt formulation in improving system reliability.

The main contributions of this work are fourfold:

- a structured architecture for AI-augmented value investing integrating extraction, modeling, explainability, and governance;
- a formalization of human-in-the-loop supervision as an operational control mechanism ensuring traceability and decision accountability;
- an operational characterization of prompt literacy as a determinant of AI system reliability and analytical correctness;
- an empirical evaluation demonstrating significant productivity gains and reliability improvements in AI-assisted financial analysis.

By linking AI performance to structured human interaction and prompt design, this work clarifies how AI can be integrated more safely and effectively into financial decision-support systems. More broadly, it suggests that future competitive advantage in AI-augmented investing will depend not only on access to data or computational resources but also on the ability to interact with AI systems in a structured, verifiable, and economically grounded manner.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on AI-assisted financial analysis, decision-support systems, explainable AI, and generative models and identifies the research gap addressed by this work. Section 3 revisits

https://doi.org/10.3390/electronics15061155

the fundamental principles of value investing and analyzes the operational and cognitive limitations of traditional analytical workflows. Section 4 introduces the proposed AI-augmented value investing architecture, describing its design objectives and the three-layer system structure integrating data extraction, valuation modeling, and explainability and governance. Section 5 details the data layer, focusing on automated financial information extraction, normalization, and integration into analyst workflows. Section 6 presents the modeling layer, including automated KPI computation, intrinsic value estimation via discounted cash flow, and NLP-assisted qualitative analysis. Section 7 discusses explainability and governance mechanisms, including traceability, model risk management, and the role of explainable AI in ensuring analytical reliability. Section 8 formalizes the human-in-the-loop operating model and introduces prompt literacy as a core operational competence enabling reliable human–AI interaction. Section 9 presents the case study using the Rivanna AI platform, including experimental design, validation metrics, and empirical results on extraction correctness, valuation consistency, and productivity gains. Finally, Section 10 concludes the paper by summarizing the main contributions, discussing limitations, and outlining directions for future research.

To improve readability and terminological consistency, the principal acronyms used throughout the manuscript are consolidated here: Artificial Intelligence (AI), Capital Asset Pricing Model (CAPM), Chain-of-Thought (CoT), Discounted Cash Flow (DCF), Decision Support System (DSS), Extract–Load–Transform (ELT), Equity Risk Premium (ERP), Extract–Transform–Load (ETL), Fiscal Year (FY), Generally Accepted Accounting Principles (GAAP), Human-in-the-Loop (HITL), International Financial Reporting Standards (IFRS), Key Performance Indicator (KPI), Large Language Model (LLM), Local Interpretable Model-Agnostic Explanations (LIME), Management's Discussion and Analysis (MD&A), Margin of Safety (MoS), Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG), Relative Error (RE), Return on Invested Capital (ROIC), U.S. Securities and Exchange Commission (SEC), Shapley Additive Explanations (SHAP), Terminal Value (TV), User Interface (UI), Weighted Average Cost of Capital (WACC), and Explainable Artificial Intelligence (XAI). Company tickers are The Coca-Cola Company (KO), PepsiCo, Inc. (PEP), and Keurig Dr Pepper Inc. (KDP).

2. Related Works

The integration of artificial intelligence (AI) into financial analysis has attracted increasing attention across multiple research domains, including financial forecasting, automated information extraction, decision-support systems, and explainable AI. This section situates the present work within the existing literature and highlights the specific contribution of our approach.

2.1. AI for Financial Analysis and Valuation

The application of computational methods to financial analysis has evolved significantly over the past decades. Classical approaches relied primarily on statistical models for valuation and forecasting, including regression-based discounted cash flow (DCF) modeling and factor-based valuation methods [1,2]. These frameworks remain foundational to value investing, which emphasizes intrinsic value estimation and margin-of-safety reasoning.

More recently, machine learning techniques have been applied to financial prediction and classification tasks, including bankruptcy prediction, earnings forecasting, and asset pricing [3,4]. Deep learning methods have further expanded these capabilities, enabling automated extraction and interpretation of financial information from large and heterogeneous datasets [5].

https://doi.org/10.3390/electronics15061155

Natural language processing (NLP) has played a particularly important role in financial analysis. Transformer-based architectures such as BERT and related models have demonstrated strong performance in extracting structured financial information from unstructured text, including regulatory filings, earnings transcripts, and financial disclosures [6,7]. Sentiment analysis and textual embeddings have been shown to improve predictive accuracy in financial forecasting and risk assessment tasks [8,9].

However, most existing studies focus on predictive modeling rather than supporting the full analytical workflow required in value investing, which involves data extraction, financial normalization, ratio computation, valuation modeling, and strategic interpretation.

2.2. Decision-Support Systems and Augmented Intelligence

Decision-support systems (DSS) have long been used to assist complex analytical and strategic decision-making in finance and management [10]. Classical DSS architectures integrate data management, analytical models, and user interfaces to support structured reasoning.

With the emergence of AI, these systems have evolved into AI-augmented decision-support platforms capable of automating data extraction, synthesis, and analysis [11]. Rather than replacing human decision-makers, these systems operate under the paradigm of augmented intelligence, where AI enhances human cognitive capacity while preserving human supervision and responsibility [12].

Human-in-the-loop (HITL) systems have emerged as a critical design paradigm for ensuring reliability and accountability in AI-assisted workflows [13]. HITL architectures incorporate human verification and oversight at key stages of the analytical process, allowing analysts to validate extracted information, correct errors, and ensure consistency with domain knowledge.

This paradigm is particularly relevant in financial analysis, where decisions require justification, traceability, and accountability.

2.3. Explainable AI and Model Governance in Finance

Explainability has become a fundamental requirement for AI systems operating in high-stakes domains such as finance. Explainable AI (XAI) techniques enable users to interpret model outputs and understand the underlying reasoning processes [14].

Several methods have been proposed to improve explainability, including feature attribution methods such as SHAP (Shapley Additive Explanations) [15] and LIME (Local Interpretable Model-Agnostic Explanations) [16]. These techniques provide insights into the relative importance of input variables and allow analysts to validate model behavior.

Explainability is particularly important in financial decision-making, where investment decisions must be justified to stakeholders and regulatory bodies. Transparent AI systems improve trust, facilitate auditing, and reduce operational risk [17].

However, most existing explainability research focuses on predictive models rather than generative AI systems used for analytical synthesis and reasoning.

2.4. Generative AI and Large Language Models in Financial Workflows

The emergence of large language models (LLMs) has significantly expanded the scope of AI-assisted analytical workflows. Transformer-based architectures introduced by Vaswani et al. [6] and subsequent models such as GPT and BERT have enabled automated document understanding, structured data extraction, and reasoning over complex textual information.

These models have demonstrated strong performance in financial document analysis, automated report generation, and information extraction tasks [7,18].

https://doi.org/10.3390/electronics15061155

However, generative AI systems introduce new operational challenges, including hallucinations, sensitivity to prompt formulation, and probabilistic output variability [19]. Unlike deterministic software systems, LLM outputs depend not only on input data but also on the structure and clarity of the prompt provided by the user.

Recent research in human–AI interaction suggests that structured prompts and task decomposition significantly improve system reliability and output quality [20]. Nevertheless, this aspect has not been formally integrated into financial decision-support frameworks.

2.5. Research Gap and Contribution of This Work

Despite significant progress in AI-assisted financial analysis, three important gaps remain.

- First, most existing work treats AI systems as autonomous analytical tools, rather than interactive systems whose reliability depends on structured human guidance.
- Second, the role of prompt formulation as a determinant of system reliability has not been formally characterized in financial analysis workflows.
- Third, there is limited empirical evidence linking human–AI interaction design to measurable improvements in analytical efficiency and reliability.

This work addresses these gaps by proposing a unified framework for AI-driven value investing that integrates automated extraction, valuation modeling, explainability, and human-in-the-loop governance, while explicitly incorporating prompt literacy as a critical operational competence.

Through an empirical case study using an AI-assisted financial analysis platform, we provide evidence that structured prompt formulation improves extraction correctness, reduces verification overhead, and enhances analytical efficiency.

This contribution advances the literature on augmented intelligence and AI-assisted financial decision support by framing financial analysis as a collaborative human–AI reasoning process rather than a fully automated pipeline.

3. Value Investing Principles and Workflow Limitations

Value investing is grounded in the principle that financial securities represent ownership claims on real economic assets and should therefore be evaluated based on intrinsic value rather than short-term market fluctuations. Intrinsic value represents an estimate of the present value of future cash flows, adjusted for risk and capital structure. Because intrinsic value estimation is inherently uncertain, value investors rely on the concept of margin of safety, defined as the difference between estimated intrinsic value and market price, which provides protection against estimation errors and adverse developments.

A central conceptual framework supporting this approach is the distinction between price and value. Market prices reflect collective expectations, sentiment, and behavioral factors, whereas intrinsic value reflects underlying economic fundamentals such as profitability, reinvestment capacity, and competitive positioning. The role of the analyst is therefore to independently estimate intrinsic value and exploit deviations between price and value when sufficient margin of safety exists.

While these principles remain conceptually unchanged, their practical implementation has become increasingly complex due to the rapid growth in financial data volume, heterogeneity of information sources, and increased reporting complexity.

Traditional Financial Analysis Workflow and Structural Bottlenecks

The traditional value investing workflow consists of five sequential stages: data acquisition, data normalization, financial ratio computation, qualitative business analysis, and valuation modeling. Analysts collect financial statements, regulatory filings, earnings transcripts, and market data, then normalize and integrate this information to compute

https://doi.org/10.3390/electronics15061155

performance indicators such as profitability, leverage, and valuation multiples. This information is subsequently used to build valuation models, typically based on discounted cash flow (DCF) analysis.

This workflow faces two fundamental limitations.

- First, operational constraints limit scalability. Data acquisition and normalization remain labor-intensive, particularly when integrating heterogeneous sources such as structured financial statements and unstructured textual disclosures. Differences in accounting standards, reporting conventions, and fiscal calendars further increase complexity.
- Second, cognitive constraints affect analytical reliability. Human analysts are subject to cognitive biases such as anchoring, confirmation bias, and selective attention, which may distort interpretation and decision-making.

These limitations create a trade-off between analytical depth and coverage, motivating the development of AI-assisted systems capable of automating repetitive tasks while preserving interpretability and analytical rigor.

4. AI-Augmented Value Investing System Architecture and Design Objectives

4.1. Design Objectives

An AI-augmented financial analysis system must satisfy several functional and operational requirements to be suitable for real-world decision support.

First, scalability is required to enable analysis of large numbers of firms without proportional increases in analyst effort. Second, evidence integrity must be preserved, ensuring that extracted information remains traceable to original sources. Third, model pluralism is necessary to integrate multiple analytical methods, including fundamental valuation models, statistical forecasting, and textual analysis.

Fourth, explainability and governance mechanisms must ensure that analytical outputs remain interpretable, verifiable, and auditable. Finally, decision orientation must be maintained, ensuring that analytical outputs support actionable strategic decisions rather than purely predictive outputs.

These requirements motivate a modular system architecture combining automated extraction, valuation modeling, explainability, and human supervision.

4.2. Three-Layer System Architecture

The proposed architecture consists of three functional layers, as shown in Figure 1: data extraction, analytical modeling, and explainability and governance. The end-to-end workflow is summarized in Algorithm 1, consistent with the three-layer architecture shown in Figure 1.

The data layer is responsible for extracting structured and unstructured financial information from heterogeneous sources, including financial statements, regulatory filings, earnings transcripts, and market data. Automated extraction techniques based on natural language processing and structured parsing enable scalable and consistent data acquisition.

The modeling layer transforms extracted data into economically meaningful analytical outputs. This includes automated computation of financial ratios, machine learning–based forecasting of key financial drivers, and intrinsic value estimation using discounted cash flow and relative valuation models.

The explainability and governance layer ensures that analytical outputs remain interpretable and verifiable. This layer provides traceability to original data sources, model interpretability mechanisms, and verification checkpoints enabling human oversight.

https://doi.org/10.3390/electronics15061155

This layered architecture enables scalable automation while preserving analytical transparency and decision accountability.

Algorithm 1 Prompt-Driven AI Value Investing Workflow with HITL Gates

Input: Analyst goal, company C, period P, constraints (units/currency, required KPIs, valuation method), source set S.

Output: Decision-support package (sheets/memo), intrinsic value band, risk flags, full traceability log.

1. **Specify task & constraints (UI)**: The analyst defines the objective, scope (C, P), required outputs (KPIs/DCF/scenarios), and constraints (units, currency, disclosure priority).
2. **Orchestrate workflow (Orchestrator)**: Load templates, route subtasks, and initialize logging/audit trail for prompts, sources, and outputs.
3. **Layer A—Data & extraction (ETL/ELT)**: Retrieve sources S; extract numeric fields with schema-constrained prompts; map entities; normalize units/scale; align fiscal periods; attach citations.
4. **G1—Data verification gate (HITL)**: Run consistency checks (units, missingness, basic accounting ties). If failures occur, re-extract with tighter constraints or escalate to analyst correction; log decisions.
5. **Layer B—Modeling & valuation**: Compute KPIs; build valuation drivers; run DCF with scenarios and sensitivity; produce relative valuation checks; generate draft synthesis.
6. **G2–G3—KPI & valuation gates (HITL)**: Validate KPI plausibility and valuation assumptions (bounds, driver sensitivity). Resolve discrepancies via source check/re-run; record adjudications.
7. **Layer C—Explainability and governance**: Generate traceability/lineage report, uncertainty notes, XAI summaries, and drift/anomaly/bias risk flags; consolidate audit logs.
8. **G4 and output delivery**: Let the analyst validate the narrative against sources, then allow them to export nonautonomous decision-support outputs (Excel-ready tables, memo, intrinsic value band, rationale, and risk flags).

https://doi.org/10.3390/electronics15061155

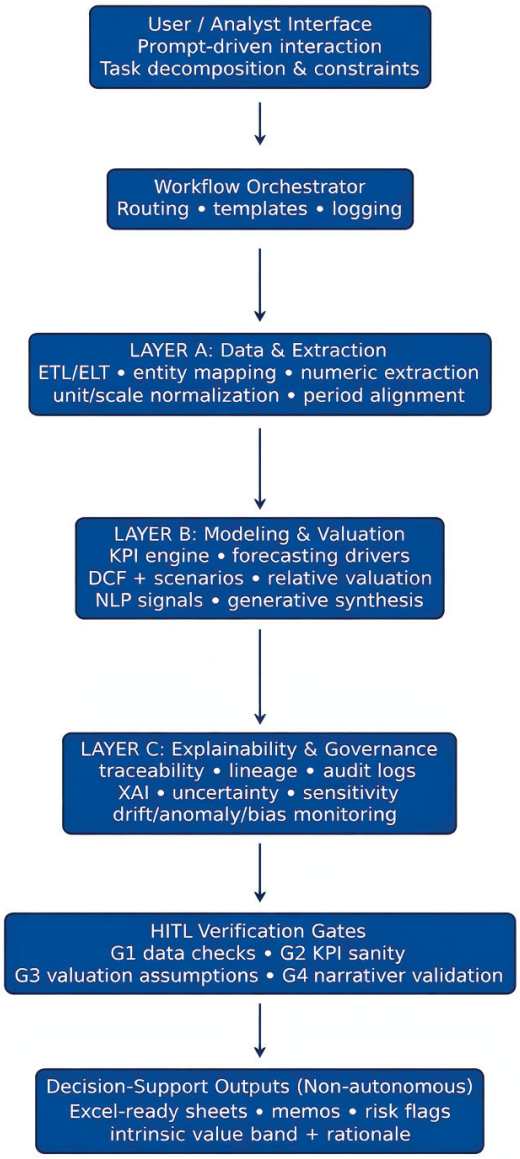


Figure 1. AI-augmented value investing architecture. The system integrates (Layer A) traceable data extraction from structured and unstructured sources, (Layer B) modeling and valuation (KPIs, forecasting of value drivers, DCF and relative valuation, NLP and generative synthesis), and (Layer C) explainability and governance (data lineage, XAI, uncertainty/sensitivity analysis, and model risk controls). Human-in-the-loop verification gates close the loop by validating data correctness, KPI consistency, valuation assumptions, and narrative outputs; the system produces decision-support artifacts rather than autonomous investment decisions.

5. Data Layer: Automated Financial Information Extraction

5.1. Automated Extraction and Normalization

The data layer is responsible for transforming heterogeneous financial information into structured, analysis-ready datasets. Modern AI techniques enable automated extraction of key financial variables from regulatory filings, earnings reports, and other disclosures. These methods support entity recognition, numerical extraction, and contextual interpretation of financial information.

Normalization mechanisms align extracted data across reporting periods, accounting conventions, and organizational structures. This process supports consistency and comparability across firms and time.

https://doi.org/10.3390/electronics15061155

A critical requirement is traceability: each extracted data point must remain linked to its original source. This enables analysts to verify correctness and supports auditability of analytical outputs.

5.2. Integration with Analyst Workflows

For practical adoption, automated extraction systems must integrate seamlessly with existing analyst workflows. Spreadsheet-based environments remain the dominant analytical interface in finance. Automated systems can generate structured tables, standardized financial metrics, and valuation-ready datasets directly within these environments.

This integration reduces manual data manipulation and allows analysts to focus on higher-level analytical tasks, including interpretation, validation, and strategic reasoning.

6. Modeling Layer: Financial Analysis and Valuation

6.1. Automated Financial Metrics and Consistency Validation

Once financial data is structured, automated systems compute financial metrics including profitability ratios, leverage indicators, liquidity measures, and valuation multiples. Automated computation supports consistency and scalability.

Validation mechanisms are required to detect anomalies such as unit mismatches, missing values, and structural inconsistencies. These mechanisms improve reliability and reduce error propagation into valuation models.

6.2. Forecasting and Intrinsic Value Estimation

Intrinsic value estimation requires forecasting future cash flows and discounting them using appropriate risk-adjusted discount rates. AI-based forecasting models can assist in estimating revenue growth, operating margins, and capital requirements.

These models should be treated as decision-support tools rather than autonomous predictors. Human analysts remain responsible for validating assumptions and ensuring consistency with economic fundamentals.

Intrinsic value estimation remains anchored in economic logic, with AI serving to improve scalability and analytical efficiency.

6.3. Natural Language Processing for Qualitative Analysis

Qualitative information plays a critical role in financial analysis. NLP techniques enable automated analysis of earnings transcripts, management communication, and financial disclosures.

These methods extract sentiment indicators, detect linguistic uncertainty, and identify changes in managerial tone or strategic direction. These signals complement quantitative analysis and support more comprehensive evaluation of firm fundamentals.

6.4. Generative AI for Analytical Synthesis

Generative AI systems assist analysts by synthesizing financial information, summarizing disclosures, and structuring analytical reports. These systems improve efficiency but introduce risks related to hallucinations and unverifiable outputs.

To mitigate these risks, generative outputs must remain traceable to source data and subject to human validation.

7. Explainability and Governance

7.1. Explainability Requirements in Financial Decision Support

Explainability is a fundamental requirement for AI-assisted financial decision support systems due to the economic, operational, and regulatory implications of investment decisions.

https://doi.org/10.3390/electronics15061155

Unlike purely predictive applications, where performance metrics alone may suffice, financial analysis requires that conclusions be interpretable, verifiable, and economically justified.

Investment decisions often undergo multiple layers of review, including analyst validation, investment committee approval, audit processes, and regulatory scrutiny. In these contexts, analytical outputs must be accompanied by clear explanations that enable decision-makers to understand how conclusions were derived and which assumptions and data inputs influenced the result.

Explainability therefore serves several critical functions. First, it enables correctness verification, allowing analysts to confirm that extracted data, computed metrics, and valuation outputs are logically and numerically consistent. Second, it supports error detection by enabling the identification of incorrect assumptions, data inconsistencies, or model failures. Third, it improves trust and usability, allowing analysts to interpret AI-generated outputs in terms of economically meaningful variables such as profitability, growth, reinvestment, and capital structure.

From a regulatory and governance perspective, explainability is also essential for auditability and accountability. Financial decision processes must maintain documented reasoning chains linking conclusions to underlying evidence. AI-assisted systems must therefore provide transparent analytical pipelines that preserve traceability from raw data to final analytical outputs.

In this context, explainability is not merely a desirable feature but a structural requirement for safe and reliable deployment of AI in financial analysis workflows.

7.2. Explainability Mechanisms

Explainability in AI-assisted financial analysis can be implemented through a combination of structural, analytical, and model-level mechanisms that ensure interpretability, traceability, and robustness.

A primary requirement is data lineage and traceability, ensuring that each extracted data point remains linked to its original source. This enables analysts to verify correctness by inspecting the originating document, such as a regulatory filing or earnings report. Traceability mechanisms typically include source references, document identifiers, and extraction metadata.

Feature attribution mechanisms provide insight into which input variables contribute most strongly to analytical outputs. In valuation models, these mechanisms allow analysts to identify key drivers such as revenue growth, operating margins, or discount rates that influence intrinsic value estimates. This improves interpretability and allows validation of economic plausibility.

Sensitivity analysis provides an additional layer of explainability by quantifying how analytical outputs change in response to variations in key assumptions. For example, intrinsic value estimates can be evaluated under alternative discount rate and growth assumptions, producing valuation ranges rather than single-point estimates. This approach reflects the inherent uncertainty of financial forecasting and improves robustness of decision support.

Uncertainty quantification mechanisms further enhance interpretability by providing confidence intervals or scenario-based output ranges. Rather than presenting deterministic predictions, explainable systems communicate uncertainty explicitly, enabling analysts to assess risk and robustness.

In generative AI components, explainability also requires source-grounded generation. Outputs such as summaries or analytical narratives must remain linked to source documents, preventing unverifiable or hallucinated conclusions. This supports the fact that narrative outputs remain evidence-based and suitable for decision support.

https://doi.org/10.3390/electronics15061155

Collectively, these mechanisms ensure that AI-assisted financial analysis remains interpretable, verifiable, and aligned with fundamental economic reasoning.

7.3. Model Risk and Governance

LLM-enabled decision-support for value investing introduces a layered risk surface across (i) source/provenance, (ii) extraction and normalization, (iii) KPI and valuation modeling, (iv) generative narrative, and (v) operations and lifecycle drift. We therefore treat governance as a system-engineering assurance layer that enforces traceability, measurable verification gates, and documented accountability across the pipeline (Algorithm 1; Figure 1), consistent with AI risk-management and lifecycle monitoring practices [21,22].

(1) **Data and source risk.** Failures arise from incomplete or stale disclosures, restatements, inconsistent accounting conventions, and mixing audited filings with lower-authority sources (decks/transcripts). The framework enforces a source-of-truth hierarchy (audited/regulatory first), timestamps all inputs, and applies completeness checks (coverage by year/statement/notes). When non-filing sources are used, items are explicitly labeled and excluded from "hard-number" computations unless independently validated. Documentation artifacts further improve auditability and reproducibility [23,24].

(2) **Extraction and normalization risk.** Common failure modes include schema drift, unit/scale errors (thousands vs. millions), fiscal-period misalignment, and synonym collisions in line items. Mitigation combines (i) schema-constrained prompts (table/JSON), (ii) mandatory reporting of units/currency/period mapping, (iii) per-number citations, and (iv) deterministic checks (range/sign constraints and cross-statement reconciliation). Any missing citation or unit ambiguity is treated as a hard failure that triggers HITL Gate G1 [25].

(3) **KPI and valuation model risk.** Even with correct inputs, valuation uncertainty is dominated by assumptions (growth, margins, reinvestment, discount rate, terminal value). To avoid overconfident point estimates, the system produces intrinsic value as a band via scenario and sensitivity grids, enforces explicit parameter bounds (e.g., terminal growth constraints), and requires analyst justification for the highest-leverage drivers at HITL Gate G3. This reflects established valuation practice that emphasizes sensitivity to key drivers and conservative decision rules (margin of safety) [2].

(4) **Generative and reasoning risk.** LLMs may produce fluent but unsupported claims ("hallucinations"), particularly in qualitative synthesis (moat, strategy, competitive dynamics) or when extrapolating beyond evidence. The framework separates numbers (must be cited and checked) from narratives (must be citation-grounded or labeled as hypotheses). Retrieval-grounded generation (retrieve-then-write) reduces unsupported statements, while structured prompting and consistency checks reduce numerical extraction errors [26–28].

(5) **Operational governance risk and verifier trust boundary.** System behavior can drift with model updates, prompt/template changes, and data shifts. Moreover, verification components can become single points of failure if treated as infallible. We therefore treat the verifier as a fallible component: every run produces an audit trail (prompts, outputs, citations, adjudications, sign-offs), critical components are version-pinned, and monitoring tracks error rates, disagreement frequency, and regression failures over time. Disagreements escalate via a documented protocol (from source check, through re-prompt, to human adjudication), and final accountability remains with HITL sign-off at G2–G4 [21,22].

Table 1 below sums up model risks, failure modes, detection signals, control, and ownership according to the five points discussed above.

https://doi.org/10.3390/electronics15061155

Table 1. Model risks, failure modes, detection signals, controls, and ownership (HITL).

Risk Category	Typical Failure Mode	Detection Signal	Primary Control/Mitigation	Owner/Gate	Residual Risk
Data & source	Missing/stale filings; missed restatement	Coverage gaps; timestamp mismatch	Source hierarchy; completeness + versioning	Analyst (G1)	Moderate
Data & source	Filing vs. deck/transcript conflict	Contradictions; cite mismatch	Prefer audited; corroborate or label	Analyst (G1/G4)	Low–Mod.
Extraction	Wrong mapping (schema drift)	Schema fail; KPI anomaly	Schema prompts; validators; retry	Analyst (G1)	Low
Extraction	Unit/scale/FX error	Out-of-range; ratio spike	Require units; checks; reconcile	Analyst (G1)	Low–Mod.
Normalization	Fiscal-period misalignment	YoY jump; totals mismatch	Period rules; reconciliation	Analyst (G1)	Moderate
KPI computation	Ratio/formula mistake	Invariant fail	Deterministic formulas; unit tests	Analyst (G2)	Low
Valuation model	Overconfident estimate; TV dominates	High sensitivity; TV share high	Scenarios; bounds; MoS band	Analyst (G3)	Moderate
Narrative	Unsupported claims	No citations; contradictions	Cite-or-label; retrieve-then-write	Analyst (G4)	Low–Mod.
Cross-model	GPT vs. Claude disagreement	Δ above threshold	Filing check; re-prompt; adjudicate	Analyst (G1/G3)	Low
Operations	Drift after updates	Regression fail; metric shift	Version pin; log changes; re-baseline	System owner	Moderate
Verifier boundary	Verifier false pass/fail	FP/FN rising	Monitor; calibrate; escalate	Sys owner + Analyst	Moderate

7.4. Disagreement Resolution and Cross-Model Validation Protocol

To improve robustness and reduce model-specific failure modes, the proposed framework includes a cross-model validation stage in which the same extraction task is independently executed by two large language models (e.g., GPT-4 and Claude). The objective is not to assume that agreement implies correctness but to use model diversity as a practical mechanism for detecting unstable outputs, unsupported values, and source inconsistencies before these propagate to KPI computation and valuation [29].

The protocol begins with parallel extraction. Given the same company, reporting period, and source set, both models receive an equivalent structured prompt requesting the extraction of predefined financial line items together with units, currency, fiscal-period mapping, and exact source citations. The outputs are then passed to a normalization layer, which harmonizes numerical scale (e.g., thousands vs. millions), currency denomination, and fiscal-period labeling and maps synonymous labels to a common schema. This step is necessary because nominal disagreements may arise from formatting or reporting conventions rather than from true content divergence [25,28].

After normalization, the system applies a disagreement detection rule. A field is flagged if at least one of the following conditions holds: (i) the relative deviation between the two extracted values exceeds a predefined threshold; (ii) the cited source passages differ materially or one output lacks a citation; (iii) the reported units, currency, or fiscal-period mapping are inconsistent; or (iv) one model returns a value while the other returns a missing field or a qualitatively different interpretation. Flagged items are collected into a disagreement set and routed to an adjudication procedure [26,27].

The adjudication hierarchy follows three ordered steps. First, the system checks the primary filing or audited report as the source of truth. If the discrepancy can be resolved directly from the filing, the verified value is accepted and logged. Second, if the source text is ambiguous or the extraction failed because of formatting complexity, the task is re-run

https://doi.org/10.3390/electronics15061155

with a tighter prompt that imposes stricter schema constraints, explicit unit handling, and a narrower extraction scope. Third, if disagreement persists, the item is escalated to a human analyst, who makes the final determination and records the rationale for the decision. In this way, unresolved conflicts are not silently absorbed into downstream computations.

A key component of this protocol is the audit trail. For each disputed item, the system stores the original prompts, both model outputs, normalized values, cited passages, discrepancy type, re-prompt attempts, and the final adjudicated value. This provides full traceability for later review and supports reproducibility, accountability, and post hoc error analysis. The protocol therefore operationalizes governance not as a generic oversight principle but as a concrete mechanism for handling uncertainty, disagreement, and model fallibility in AI-assisted financial analysis [21,22,24].

It is important to note that, beyond error detection on quantitative fields, using multiple LLMs enables what we term a *scenario corridor* for qualitative and strategic drivers. For numerical extraction, the goal of cross-model validation is convergence to a single source-grounded value after normalization and citation checks. In contrast, for qualitative judgments (e.g., competitive positioning, moat sustainability, management execution, regulatory exposure), the objective is not strict convergence to one "true" statement but the identification of a bounded set of plausible outcomes. Agreement across independent models on the *direction* and *range* of qualitative conclusions provides a more robust representation of uncertainty than relying on a single model's narrative. In the proposed governance protocol, this corridor is treated as an input to the analyst's decision-making: convergent qualitative conclusions increase confidence, while divergent outputs trigger targeted evidence retrieval and human adjudication before final sign-off.

8. Human-in-the-Loop Operating Model

8.1. Role of Human Supervision

Human-in-the-loop (HITL) supervision represents a foundational component of AI-augmented financial analysis systems. Rather than fully automating analytical workflows, HITL architectures integrate human verification and oversight at critical stages of the analytical pipeline, ensuring correctness, interpretability, and accountability.

Financial analysis involves high-stakes decisions where small numerical or logical errors may propagate into significant valuation inaccuracies. Automated extraction, modeling, and generative reasoning systems operate probabilistically and may introduce errors related to data extraction, model assumptions, or reasoning inconsistencies. HITL supervision mitigates these risks by incorporating structured human validation at predefined verification gates.

Human supervision performs several essential functions. First, analysts validate extracted data by checking unit consistency, accounting identity relationships, and source traceability. Second, analysts review model outputs to ensure economic plausibility and consistency with business fundamentals. Third, analysts verify generative outputs, ensuring that summaries, analytical narratives, and valuation conclusions remain grounded in verifiable evidence.

From a systems perspective, HITL supervision acts as a reliability layer. Its purpose is to detect and correct errors before they affect downstream analytical outputs. This approach enables scalable automation while preserving decision integrity and accountability.

HITL supervision does not eliminate efficiency gains. Instead, it shifts analyst effort away from repetitive tasks and toward validation, adjudication, and interpretation. This results in a system that combines the scalability of automation with the reliability of expert human judgment.

https://doi.org/10.3390/electronics15061155

8.2. The Augmented Analyst: From Manual Processing to Strategic Oversight

The integration of AI systems fundamentally transforms the role of financial analysts. In traditional workflows, analysts devote substantial time to manual data collection, normalization, spreadsheet manipulation, and mechanical ratio computation. These tasks are necessary but do not directly contribute to strategic insight or decision quality.

AI-assisted systems automate these operational tasks, allowing analysts to focus on higher-level cognitive functions. The role of the analyst evolves from manual processor to strategic supervisor of analytical workflows. Analysts now focus on defining analytical objectives, validating model outputs, interpreting valuation results, and integrating analytical conclusions into coherent investment narratives.

This transformation introduces a shift from procedural execution to supervisory control. Analysts no longer perform every analytical operation manually but instead oversee automated processes, ensuring correctness, consistency, and economic validity.

The augmented analyst performs several key supervisory functions:

- defining analytical objectives and problem formulation;
- validating extracted data and resolving inconsistencies;
- evaluating model assumptions and forecasting plausibility;
- interpreting valuation outputs and sensitivity analysis;
- identifying anomalies and potential model failures;
- integrating quantitative and qualitative information into investment decisions.

This supervisory role preserves human responsibility and accountability while leveraging the computational capabilities of AI systems.

Importantly, AI systems do not replace financial expertise but amplify it. Economic reasoning, domain knowledge, and judgment remain essential for interpreting model outputs and ensuring analytical correctness.

8.3. Prompt Literacy as a Core Operational Competence

A critical skill emerging in AI-augmented financial analysis is prompt literacy, defined as the ability to formulate structured, precise, and context-aware instructions that guide AI systems toward reliable, verifiable, and decision-relevant outputs.

Unlike deterministic software, generative AI models respond probabilistically to the input they receive. The prompt therefore acts as an operational control signal. Its formulation influences extraction accuracy, analytical consistency, and interpretability.

Poorly specified prompts may lead to incomplete extraction, incorrect assumptions, logical inconsistencies, or hallucinated outputs. These failures increase verification overhead and reduce system reliability. Conversely, structured and precise prompts improve correctness, reproducibility, and analytical efficiency.

From an operational standpoint, effective prompts incorporate four essential components.

First, explicit task definition clearly specifies the analytical objective, such as extracting specific financial variables, computing defined metrics, or performing structured valuation analysis. Precise task specification reduces ambiguity and improves extraction accuracy.

Second, contextual constraints provide necessary background information, including company identity, reporting period, accounting standards, and unit conventions. Context reduces uncertainty and ensures consistency with analytical requirements.

Third, structured output specification defines the required output format, such as tabular data, spreadsheet-compatible values, or predefined analytical structures. Structured outputs improve interpretability and facilitate downstream processing.

Fourth, verification orientation explicitly requires source traceability, citations, and uncertainty indicators. This ensures that outputs remain verifiable and suitable for financial decision support.

https://doi.org/10.3390/electronics15061155

In addition to prompt structure, task decomposition plays an important role in improving system reliability. Complex analytical tasks can be decomposed into smaller, sequential subtasks, allowing incremental validation and reducing error propagation. This approach improves robustness and enables systematic verification of intermediate outputs.

From a systems perspective, prompt formulation is an interaction-level control mechanism. Prompt literacy can therefore be viewed as an operational competence, similar in importance to model calibration or parameter tuning in traditional analytical systems.

This insight introduces an important conceptual shift: the reliability of AI-assisted financial analysis depends not only on model architecture and training data but also on the quality of human–AI interaction. The analyst actively shapes the reasoning trajectory of the AI system through structured prompt design.

In this sense, prompt literacy becomes a functional component of analytical rigor, ensuring that AI-assisted systems operate as reliable decision-support tools rather than autonomous decision-makers.

8.4. Closed-Loop Human–AI Interaction and Reliability Control

The HITL operating model establishes a closed-loop interaction between human analysts and automated analytical systems. In this loop, AI systems generate structured analytical outputs, which are then validated, refined, and corrected by human supervisors. These corrections are incorporated into subsequent analytical iterations through prompt refinement and workflow adjustment.

This closed-loop structure improves system reliability through iterative error correction and continuous validation. Human supervision ensures that system outputs remain aligned with economic reality and investment logic.

Over time, this interaction produces a stable and reliable analytical process in which AI systems provide scalable computational support while human analysts maintain supervisory control.

This hybrid human–AI architecture represents a practical and robust approach to integrating AI into financial decision-making, combining computational scalability with human judgment, accountability, and interpretability.

9. Case Study: Competitive Analysis with an AI Platform (Rivanna AI)

9.1. Use Case and Scope

To validate the proposed AI-augmented value investing framework, we conducted a case study using Rivanna AI [30], an AI-assisted financial analysis platform, applied to a comparative analysis of three publicly traded companies in the U.S. beverage sector: The Coca-Cola Company (KO), PepsiCo Inc. (PEP), and Keurig Dr Pepper Inc. (KDP).

This sector provides a suitable evaluation environment due to its well-documented financial disclosures, comparable business models, and stable reporting practices. The focus on U.S.-listed firms supports high-quality and standardized primary data sources, including SEC filings (10-K, 10-Q, and 8-K), earnings transcripts, and financial statements, enabling rigorous evaluation of extraction accuracy, valuation consistency, and workflow efficiency.

The objective of the case study is to evaluate whether an AI-augmented, human-in-the-loop (HITL) analytical pipeline can improve the scalability and efficiency of financial analysis while preserving correctness, interpretability, and decision reliability.

9.2. Problem Formulation, Protocol, and Methods

Fundamental financial analysis requires integrating heterogeneous information sources, including structured financial statements, unstructured regulatory disclosures, and

https://doi.org/10.3390/electronics15061155

market data. Analysts must extract and normalize financial variables, compute performance indicators, and construct valuation models such as discounted cash flow (DCF) analysis.

This process is time-intensive and prone to operational and cognitive failure modes. Operational challenges include unit inconsistencies, period misalignment, and data normalization complexity. Cognitive challenges include interpretation bias and inconsistency in analytical assumptions.

As a result, analysts face a structural trade-off between analytical depth and coverage. Increasing analytical coverage across multiple firms often reduces the time available for verification and interpretation, potentially degrading analytical quality.

While AI systems offer the potential to automate extraction and synthesis, financial workflows impose stringent requirements for traceability, correctness, and auditability. Even small numerical errors in extracted inputs may propagate into significant valuation deviations. Furthermore, generative AI outputs may introduce unsupported or unverifiable analytical claims if not properly constrained.

The key challenge is therefore not merely automating analysis but designing an AI-assisted analytical system that maintains correctness, transparency, and verifiability under real-world decision-making constraints.

Rivanna AI operates as an AI-augmented decision-support system rather than an autonomous decision-making agent. Its role is to automate operational components of the financial analysis pipeline while preserving human oversight and accountability.

The system performs several analytical functions, including automated extraction of financial variables from regulatory filings, structured generation of spreadsheet-compatible financial datasets, automated computation of financial metrics, and support for valuation modeling.

However, all outputs are subject to human verification through explicit HITL validation gates. Analysts validate extracted financial values, confirm unit and period consistency, reconcile accounting identities, and review valuation assumptions before analytical outputs are used for decision-making.

An important operational observation emerging from the case study concerns the dependence of system performance on prompt formulation quality. When tasks were requested with insufficient context or as broad, open-ended analytical objectives, the system exhibited higher rates of inconsistency, incomplete extraction, or structurally incorrect outputs, increasing verification time and cognitive load for the analyst. Conversely, when the same analysis was decomposed into smaller, explicitly defined subtasks—with clear scope, period specification, and output format requirements—the reliability and consistency of the system improved significantly.

This behavior reflects a fundamental property of probabilistic generative systems: their outputs are conditioned not only on training data but also on the structure of the prompt itself. As a result, prompt formulation becomes an operational control mechanism that directly influences system correctness, efficiency, and usability. In this sense, prompt literacy is not merely a user convenience, but an integral component of the AI-supported analytical workflow.

The evaluation pipeline consisted of four sequential analytical tasks:

1. Extraction and normalization of fundamental financial variables from regulatory filings;
2. Computation of key performance indicators, including profitability, leverage, and valuation multiples;
3. Intrinsic value estimation using discounted cash flow (DCF) modeling;
4. Generation of structured analytical summaries and investment rationale.

https://doi.org/10.3390/electronics15061155

The intrinsic value was estimated using a standard DCF framework:

$$V = \sum_{t=1}^{n} \frac{FCF_t}{(1 + WACC)^t} + \frac{TV}{(1 + WACC)^n}$$

where the terminal value is defined using the Gordon growth model:

$$TV = \frac{FCF_{n+1}}{WACC - g}$$

DCF sensitivity analysis was performed across multiple discount rate and terminal growth assumptions to evaluate valuation robustness.

Prompts were designed to impose explicit extraction constraints, including fixed output schemas, unit and fiscal-period reporting, and source traceability requirements. To make this protocol more concrete, Box 1 reports an example of the structured prompt used in the case study for financial-variable extraction and numerical verification.

Box 1. Example of structured prompt for financial extraction and numerical verification.

Task: Extract the following financial variables for **[Company]** for fiscal year **[FY2024]** from the provided primary filing.
Variables required: Revenue, Operating Income, Net Income, Total Assets, Total Liabilities, Shareholders' Equity, Capital Expenditures, Depreciation and Amortization, Cash and Cash Equivalents, and Total Debt.
Instructions:

1. Use **only the provided primary filing** as the source of truth. Do not use prior knowledge or external information.
2. Return the results in a **structured table** with exactly the following columns:
   - Variable
   - Reported value
   - Unit (units/thousands/millions)
   - Currency
   - Fiscal period
   - Source citation
   - Notes/flags
3. Preserve the **exact reported numerical values** as stated in the filing. Do not round unless the source itself is rounded.
4. If a value is reported under a different but equivalent label, extract it and indicate the mapping in **Notes/flags**.
5. If a value is missing, ambiguous, or not directly stated, return **"NOT FOUND"** and explain briefly in **Notes/flags**.
6. Before returning the table, perform the following **self-checks**:
   - ○ verify that all values use consistent units and currency;
   - ○ verify that fiscal-period labels are aligned;
   - ○ check whether the accounting identity **Assets ≈ Liabilities + Equity** is approximately satisfied when the relevant fields are available;
   - ○ flag any likely unit mismatch, missing value, inconsistency, or unsupported citation.
7. If the cited passage does not fully support the extracted value, return **"REVIEW REQUIRED"** in **Notes/flags**.
8. Do **not** provide free-form reasoning or narrative explanation.

Output format:
Return only:
(a) the structured table; and
(b) a short bullet list titled **Detected Issues**, reporting any inconsistencies or flags.

This structured pipeline enables evaluation of both analytical correctness and operational efficiency.

The experimental evaluation addresses three research questions:

https://doi.org/10.3390/electronics15061155

- RQ1: Extraction correctness—How accurately does the system extract financial variables compared to manually verified ground truth?
- RQ2: Analytical consistency—Do AI-assisted KPI and valuation outputs remain consistent with manually verified analysis?
- RQ3: Productivity impact—How does AI assistance affect total analytical time and task allocation?

We report four extraction metrics.

1. Exact-match accuracy = correctly extracted fields/total evaluated fields.
2. Relative error (RE) = $\frac{|x_{AI} - x_{ref}|}{|x_{ref}| \times 100}$.
3. Unit/scale error rate = fields requiring unit-scale correction/total fields.
4. Missingness = unavailable extracted fields/total requested fields.

For KPI consistency, mismatch rate is computed as inconsistent KPIs over total KPIs per family.

All percentages are reported with explicit denominators in the corresponding tables.

In particular, extraction accuracy was evaluated using exact-match accuracy, relative error, unit consistency, and missing data rates.

Analytical consistency was evaluated by comparing computed KPIs and intrinsic valuation estimates with manually verified reference values.

Productivity impact was measured by comparing total workflow time between manual and AI-assisted analysis, including extraction, modeling, and validation tasks.

9.3. Results

The experimental results provide evidence that AI-assisted analysis maintains high extraction accuracy, with an overall exact-match rate of 87.5% and negligible unit consistency errors. Relative extraction errors remained below 1% across evaluated financial variables, indicating high numerical reliability.

KPI computations and valuation outputs remained consistent with manually verified reference results, confirming analytical correctness of AI-assisted workflows when combined with HITL validation.

Table 2 reports extraction correctness, through accuracy, relative error, scale error, missingness per company. Table 3 shows KPI consistency, via mismatch rate per KPI family (profitability, leverage, multiples). Table 4 reports the time study, that is, the time comparison and breakdown between a completely manual process and the Rivanna + HITL process (mean $\pm$ std). Finally, Table 5 reports the HITL ablation, showing the impact of verification gates on the critical-error rate.

Table 2. Extraction correctness for the FY2024 micro-benchmark (KO, PEP, KDP).

Company	Period(s)	# Fields (F)	Exact-Match (%)	Median RE (%)	Mean RE (%)	Scale/Unit Err. (%)	Missingness (%)
KO	FY2024	2	50.0	0.783	0.783	0.0	0.0
PEP	FY2024	4	100.0	0.144	0.171	0.0	0.0
KDP	FY2024	2	100.0	0.325	0.325	0.0	0.0
All	FY2024	**8**	**87.5**	**0.261**	**0.362**	**0.0**	

Notes: Exact match is computed with a 1% relative-error tolerance. RE = relative error. Scale/unit error rate measures million/billion (or unit) mismatches. Missingness is the fraction of unavailable extracted fields among the selected benchmark fields.

https://doi.org/10.3390/electronics15061155

Table 3. KPI consistency and valuation metrics (FY2024) for KO, PEP, and KDP.

KPI (FY2024)	KO	PEP	KDP
Operating margin (%)	21.23	14.03	16.88
Net margin (%)	22.63	9.82	9.39
P/E (x)	25.31	21.87	30.48
P/B (x)	10.79	11.57	1.79
EV/EBITDA (x)	27.29	15.24	17.59
FCF yield (%)	1.85	3.61	3.31

Notes: Operating margin = EBIT/Revenues. Net margin = Net income/Revenues. P/E = Price/Diluted EPS. P/B = Price/(Equity per share). EV/EBITDA = (Market Cap + Total Debt − Cash)/EBITDA. FCF yield = (Operating Cash Flow − Capex)/Market Cap. Values are rounded to two decimals and aligned to FY2024 inputs used in the case study.

Table 4. DCF sensitivity and price–value gap stability (FY2024).

Company	Market Price (End-2024)	Base Intrinsic Value	Min Intrinsic (WACC +1%, g −0.5%)	Max Intrinsic (WACC −1%, g +0.5%)	Gap Sign Stable? (Y/N)	Ranking Stable? (Y/N)
KO	62.0	68.0	55.0	96.6	N	N
PEP	152.0	136.0	110.0	193.1	N	N
KDP	32.0	36.0	29.1	51.1	N	N

Notes: Base intrinsic values come from the case-study DCF setup. Sensitivity bounds are computed with a standard DCF stress on discounting and terminal growth. Min intrinsic: higher discount rate and lower terminal growth (WACC +1%, g −0.5%). Max intrinsic: lower discount rate and higher terminal growth (WACC −1%, g +0.5%). Results are rounded to one decimal place. "Gap sign stable" indicates whether (Intrinsic–Market) keeps the same sign across min/base/max scenarios. "Ranking stable" indicates whether the relative attractiveness ranking remains unchanged across the sensitivity range.

Table 5. Time study and HITL overhead (manual vs. AI-assisted workflow).

Workflow	T1 Extract (h)	T2 KPIs (h)	T3 DCF (h)	T4 Memo Synthesis (h)	Verification Gates (h)	Total (h)	Saving vs. Manual (%)
Manual baseline	10.0	6.0	10.0	6.0	—	32.0	—
Rivanna + HITL	2.0	1.5	2.5	0.8	3.2	10.0	68.8
Reported range (case study)	—	—	—	—	Included	8–12	vs. 25–40

Notes: The first two rows provide a consistent central decomposition aligned with the reported case-study envelope.

The most significant improvement was observed in operational efficiency. Total analytical time was reduced from approximately 25–40 h using traditional manual workflows to approximately 8–12 h using AI-assisted analysis, corresponding to an efficiency improvement exceeding 65%.

Importantly, this efficiency gain did not eliminate human involvement but reallocated analyst effort from manual data processing to validation and interpretation tasks.

In particular, with respect to Table 5, the 32 h vs. 10 h values represent a central task-level decomposition, while the 25–40 h vs. 8–12 h values represent the empirical envelope observed across runs. Both are reported to distinguish structured baseline accounting from observed operational variability.

Overall effort reduction is computed as:

$$\text{Saving} = \frac{32.0 - 10.0}{32.0} \times 100 = 68.8\%$$

In the AI-assisted workflow, the reported verification gates comprise citation/source validation, unit and fiscal-period reconciliation, accounting-identity and consistency checks, and analyst review of the most influential valuation assumptions before approval of the final output.

https://doi.org/10.3390/electronics15061155

The third row reports the empirical envelope already discussed in the manuscript (manual: 25–40 h; AI-assisted: 8–12 h).

9.4. Discussion and Operational Implications

DCF analysis identified moderate valuation discrepancies across evaluated firms. Coca-Cola and Keurig Dr Pepper exhibited positive intrinsic value gaps relative to market prices, suggesting potential undervaluation under baseline assumptions. PepsiCo exhibited a negative intrinsic value gap, indicating limited margin of safety.

Sensitivity analysis confirmed that valuation conclusions remain dependent on discount rate and growth assumptions, reinforcing the importance of uncertainty-aware valuation interpretation.

These results provide evidence that AI-assisted analysis can produce economically meaningful valuation outputs suitable for decision support, provided that human validation and interpretability mechanisms are maintained.

The primary operational benefit of AI-assisted analysis lies in the automation of data extraction, normalization, spreadsheet preparation, and preliminary KPI/valuation modeling, which substantially reduces the time traditionally spent on repetitive manual information processing. In the manual workflow, analysts must search across filings, locate relevant passages, transcribe numerical values, align fiscal periods and units, populate spreadsheets, and only then begin higher-level interpretation. In the AI-assisted workflow, much of this mechanical burden is compressed by prompt-driven extraction and structured output generation.

Importantly, the observed reduction in total analysis time, from approximately 25–40 h to 8–12 h, does not imply the removal of human effort. Rather, it reflects a shift in analyst effort from low-value procedural tasks toward higher-value supervisory tasks. In the proposed framework, human work is concentrated in the verification gates: checking source citations, reconciling units and fiscal periods, resolving discrepancies between extracted values and primary filings, reviewing accounting consistency, and adjudicating high-leverage valuation assumptions before final reporting. In this sense, AI changes the composition of cognitive work rather than eliminating it.

This distinction is important for interpreting productivity gains correctly. Verification is cognitively different from original manual research. Manual research is dominated by search, transcription, and spreadsheet construction; verification is dominated by focused review, exception handling, and judgment under traceability constraints. Because the AI system pre-structures the evidence and surfaces candidate values, the analyst can allocate more attention to economically meaningful questions—whether a number is supported by the filing, whether a discrepancy is material, and whether the valuation assumptions remain plausible—rather than to clerical processing. Accordingly, the efficiency gain should be understood not as "analysis without humans", but as analysis with human effort reallocated from information processing to validation, adjudication, and interpretation.

From a systems perspective, this redistribution of effort is a central advantage of HITL architectures. Automation improves scalability, while structured verification preserves analytical reliability and decision accountability. The resulting workflow is therefore more suitable for repeatable financial decision support than either a purely manual process or an ungoverned autonomous AI pipeline.

9.5. Limitations, Threats to Validity, and Reproducibility Notes

This study has four main limitations.

- First, valuation outputs remain assumption-sensitive (discount rates, growth, and terminal value settings).

- Second, the empirical evaluation is intentionally narrow (three firms, one sector), so external validity is limited.
- Third, performance depends on data availability and document quality.
- Fourth, prompt quality and operator expertise can influence outcomes, introducing interaction-related variance.

Accordingly, results should be interpreted as case-study evidence supporting the feasibility of AI-augmented, HITL-governed workflows, rather than universal performance claims.

The case study provides evidence that AI-assisted analytical architectures can improve efficiency and scalability of financial analysis while maintaining correctness and interpretability.

Rather than replacing human analysts, AI systems function as computational accelerators within human-supervised analytical pipelines. The combination of automated extraction, explainable modeling, and human validation provides a robust framework for scalable and reliable financial decision support.

These findings support the viability of AI-augmented, human-in-the-loop architectures for real-world financial analysis applications.

An additional threat to external validity concerns the information richness of the empirical setting. The present case study is conducted on large, well-covered U.S. firms characterized by relatively standardized disclosures, high reporting frequency, and broad availability of structured and semi-structured financial information. These conditions are favorable for prompt-driven extraction, citation grounding, and downstream valuation consistency. By contrast, in small-cap firms, emerging-market issuers, or other information-scarce environments, the proposed framework may exhibit lower reliability due to thinner disclosures, more heterogeneous reporting conventions, lower-quality source formatting, and weaker representation of firm-specific patterns in the pretraining corpus of large language models. Under such conditions, one may reasonably expect higher missingness, more frequent citation ambiguity, greater disagreement across model outputs, and a larger burden of Human-in-the-Loop adjudication. Therefore, the performance gains documented in this study should be interpreted as conditional on a relatively high-information regime, and future validation should assess the robustness of the framework across firms and markets with lower disclosure quality and reduced informational coverage.

A related qualification concerns the different disclosure depth and approach typically observed across large-cap, mid-cap, and small-cap firms. Smaller issuers often provide less extensive, less structured, and less frequent public disclosure than large firms, not necessarily because fewer strategic initiatives are undertaken, but because communication resources, reporting practices, and regulatory environments may differ substantially across firms and jurisdictions. As a result, large firms are often more suitable for traceable extraction, reconciliation, and valuation support, particularly when disclosure requirements are detailed and standardized. However, this limitation does not affect AI alone; it also constrains the work of the human analyst, whose judgment is equally bounded by the quality, granularity, and comparability of the available information. In this sense, the proposed framework should be viewed as largely neutral with respect to disclosure scarcity: it does not remove differences in disclosure depth, but it provides a governed and traceable method for extracting, validating, and reasoning over whatever information is actually disclosed.

To improve reproducibility, we specify the operational protocol used in the case study.

- Scope and sources: three U.S. listed beverage firms (KO, PEP, KDP), FY2024-oriented comparative analysis based on publicly available filings and company disclosures used in the workflow.

- Task decomposition: each run follows four sequential tasks: extraction/normalization, KPI computation, DCF valuation, and memo synthesis.
- Prompt protocol: prompts are structured with explicit objectives, context constraints (firm, period, units), output schemas (tabular/spreadsheet-ready), and verification requests (source traceability and uncertainty flags).
- Ground-truth and adjudication: extracted values are compared against manually verified references; disagreements are resolved through analyst review and source-level reconciliation.
- Validation gates: HITL checks include unit/period consistency, accounting identity reconciliation, and DCF assumption review before final reporting.

This protocol is intended to make the experimental pipeline auditable and repeatable, while acknowledging that broader multi-sector validation remains future work.

10. Conclusions and Future Work

This article presented a structured framework for AI-driven value investing that combines automated data extraction, scalable modeling, explainability, and human-in-the-loop governance to support strategic financial decision-making.

The results provide case-study evidence that AI-assisted workflows can materially improve analytical productivity (from a manual range of 25–40 h to 8–12 h in our setting) while maintaining accountability through traceability and HITL verification gates. It is important to note that the quantified savings reported in this study reflect the initial implementation stage; further productivity improvements are expected through iterative prompt refinement, HITL workflow standardization, and organizational learning-curve effects beyond the currently measured values.

A key conceptual contribution of this work is the identification of prompt literacy as a core operational competence of the augmented analyst. The case study shows that the clarity, structure, and contextual completeness of prompts directly influence extraction correctness, analytical consistency, and workflow efficiency. Structured, decomposed prompts improve reliability and reduce verification overhead, whereas vague or overly complex requests increase error rates and human correction effort.

These findings reinforce a broader conclusion: AI in financial analysis functions not as an autonomous decision-maker but as a probabilistic reasoning amplifier whose effectiveness depends on structured human guidance, verification discipline, and critical interpretation.

In the long run, competitive advantage in AI-augmented investing may depend less on access to raw data or computational power, and more on the ability to interact with AI systems effectively, namely formulating precise questions, validating outputs rigorously, and integrating AI-generated insights into coherent economic reasoning.

Future research directions include:

- formal quantification of prompt quality and its impact on extraction and valuation accuracy,
- integration of probabilistic uncertainty bounds into valuation outputs,
- extension of AI-augmented workflows across broader sectors and international markets,
- development of standardized governance and verification protocols for AI-supported financial analysis.

A key next step is to validate the framework across multiple sectors, small- and mid-cap firms, and international regulatory environments, where disclosure quality, standardization, and comparability may differ substantially. Such validation is important not because information scarcity is uniquely problematic for AI, but because it represents a structural constraint on both human and AI-assisted analysis, which the proposed

https://doi.org/10.3390/electronics15061155

framework addresses through traceability and governance rather than by assuming richer information than the market actually discloses.

Ultimately, AI shifts the analyst's role from information processor to reasoning architect, namely designing, guiding, and validating analytical workflows that combine human judgment and machine intelligence.

Author Contributions: Conceptualization, M.G.; Methodology, A.C., M.G. and L.R.C.; Formal analysis, A.C.; Investigation, A.C.; Writing—original draft, A.C. and L.R.C.; Writing—review & editing, L.R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Graham, B.; Dodd, D. *Security Analysis*; McGraw-Hill: New York, NY, USA, 1934.
2. Damodaran, A. *Investment Valuation: Tools and Techniques for Determining the Value of Any Asset*; Wiley: New York, NY, USA, 2012.
3. Altman, E.I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J. Financ.* **1968**, *23*, 589–609. [CrossRef]
4. Gu, S.; Kelly, B.; Xiu, D. Empirical asset pricing via machine learning. *Rev. Financ. Stud.* **2020**, *33*, 2223–2273. [CrossRef]
5. Heaton, J.; Polson, N.; Witte, J. Deep learning in finance. *Annu. Rev. Financ. Econ.* **2017**, *11*, 1–23.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**. [CrossRef]
7. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019. [CrossRef]
8. Tetlock, P. Giving content to investor sentiment. *J. Financ.* **2007**, *62*, 1139–1168. [CrossRef]
9. Loughran, T.; McDonald, B. When is a liability not a liability? Textual analysis, dictionaries, and financial reporting. *J. Financ.* **2011**, *66*, 35–65. [CrossRef]
10. Power, D.J. *Decision Support Systems: Concepts and Resources*; Business Expert Press: New York, NY, USA, 2002.
11. Shrestha, Y.R.; Ben-Menahem, S.M.; von Krogh, G. Organizational decision-making structures in the age of artificial intelligence. *Calif. Manag. Rev.* **2019**, *61*, 66–83. [CrossRef]
12. Davenport, T.H.; Kirby, J. *Only Humans Need Apply*; Harper Business: New York, NY, USA, 2016.
13. Amershi, S.; Weld, D.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; Horvitz, E.; Iqbal, S.; Suh, J.; Teevan, J.; et al. Guidelines for human-AI interaction. In *CHI Conference on Human Factors in Computing Systems*; ACM: New York, NY, USA, 2019.
14. Doshi-Velez, F.; Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv* **2017**, arXiv:1702.08608. [CrossRef]
15. Lundberg, S.; Lee, S.I. A unified approach to interpreting model predictions. In *NeurIPS*; Curran Associates Inc.: Red Hook, NY, USA, 2017.
16. Ribeiro, M.T.; Singh, S.; Guestrin, C. Why should I trust you? Explaining the predictions of any classifier. In *KDD*; ACM: New York, NY, USA, 2016.
17. Molnar, C. *Interpretable Machine Learning*; Lulu: Raleigh, NC, USA, 2020.
18. Araci, D. FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv* **2020**, arXiv:1908.10063.
19. Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y.T.; Li, Y.; Lundberg, S.; et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv* **2023**, arXiv:2303.12712. [CrossRef]
20. Sahoo, P.; Singh, A.K.; Saha, S.; Jain, V.; Mondal, S.; Chadha, A. A Systematic Survey of Prompt Engineering in Large Language Models. *arXiv* **2025**, arXiv:2402.0792.
21. National Institute of Standards and Technology (NIST). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; NIST: Gaithersburg, MD, USA, 2023.
22. *ISO/IEC 23894:2023*; Information Technology—Artificial Intelligence—Guidance on Risk Management. International Organization for Standardization (ISO): Geneva, Switzerland; International Electrotechnical Commission (IEC): Geneva, Switzerland, 2023.
23. Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J.W.; Wallach, H.; Daumé, H., III; Crawford, K. Datasheets for Datasets. In *Communications of the ACM*; ACM: New York, NY, USA, 2021; Volume 64, pp. 86–92.

24. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I.D.; Gebru, T. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*; ACM: New York, NY, USA, 2019; pp. 220–229.
25. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-T.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates Inc.: Red Hook, NY, USA, 2020.
26. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Adv. Neural Inf. Process. Syst.* **2022**, *55*, 1–38.
27. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Zhou, D. Self-Consistency Improves Chain-of-Thought Reasoning in Language Models. *arXiv* **2022**, arXiv:2203.11171.
28. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. In *ACM Computing Surveys*; ACM: New York, NY, USA, 2023; Volume 55, pp. 1–35.
29. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. In *ACM Computing Surveys*; ACM: New York, NY, USA, 2023.
30. Rivanna AI. Available online: https://www.rivanna.ai/ (accessed on 16 February 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Retrieval-Augmented Large Language Model for Clinical Decision Support with a Medical Knowledge Graph

Fatima Saidu and Julie Wall *

School of Computing and Engineering, University of West London, London W5 5RF, UK;
32147050@student.uwl.ac.uk
* Correspondence: julie.wall@uwl.ac.uk

Abstract

This study examines clinician interactions with a Knowledge Graph (KG)-enhanced Large Language Model (LLM) for diagnostic support, with an emphasis on the rare condition pseudohypoparathyroidism (PHP). Ten medical professionals engaged with simulated diagnostic scenarios, using the KG-enhanced LLM to support reasoning and validate differential diagnoses. Evaluation included basic model performance (RAGAS = 0.85; F1 = 0.79) and clinician-centered outcomes, such as diagnostic conclusions, confidence, adherence, and efficiency. Results show the tool was most valuable for rare or uncertain cases, increasing clinician confidence and supporting reasoning, while familiar cases elicited selective adoption with minimal AI engagement. Participant feedback indicated generally high usability, accuracy, and relevance, with most reporting improved efficiency and trust. Statistical analysis confirmed that AI assistance significantly reduced time-to-diagnosis ($t(8) = 4.99$, $p = 0.001$, Cohen's $d_z = 1.66$, 95% CI [73.8, 197.2]; Wilcoxon $W = 0.0$, $p = 0.0039$). These findings suggest that KG-enhanced LLMs can effectively augment clinician judgment in complex cases, serving as reasoning aids or educational tools while preserving clinician control over decision-making. The study emphasizes evaluating AI not only for accuracy, but also for practical utility and integration into real-world clinical workflows.

Keywords: knowledge graph; large language model; retrieval-augmented generation; clinical decision support system; diagnostic accuracy; explainable AI; pseudohypoparathyroidism

1. Introduction

Healthcare systems continue to face persistent challenges due to physician shortages, increasing workloads, and high rates of stress and burnout [1]. Clinical decision support systems (CDSS), which are designed to improve healthcare outcomes, also help mitigate these pressures by providing clinicians with relevant, evidence-based guidance to inform their decisions [2]. Large Language Models (LLMs) show promise in supporting diagnostic reasoning and reducing errors [3,4]. However, LLMs often struggle with factual accuracy and long-term information retention [5,6]. They can also produce misleading or fabricated content, a limitation known as hallucination and their outputs may lack transparency, reliability, and alignment with domain-specific standards [7].

To address these limitations, a promising strategy is to enhance LLMs with Knowledge Graphs (KGs). KGs are structured representations that explicitly encode entities and the relationships between them, facilitating organised data management and supporting domain conceptualisation [8]. By grounding LLMs in this structured knowledge, KGs

improve factual accuracy, interpretability, and domain relevance [9]. Current studies [10,11] have demonstrated that integrating KGs can enhance LLM performance in diagnostic support. However, evaluations often focus primarily on performance metrics such as accuracy, without considering whether these improvements address clinician concerns or support their reasoning.

In practice, the benefits of CDSSs are frequently limited by low adoption and inconsistent use among clinicians [12]. Key factors influencing adoption include the perceived usefulness and relevance of the information, as well as the system's ease of use and efficiency [13]. Clinicians are more likely to engage with support tools in situations of high diagnostic uncertainty, such as rare or atypical cases, whereas familiar cases may evoke a low perceived need for assistance. This highlights the importance of evaluating not only model performance but also understanding how these models address clinician concerns and actively support clinical reasoning across varying diagnostic scenarios.

This study examines how clinicians interact with a KG-enhanced LLM for diagnostic support when presented with rare case presentations that can be easily misdiagnosed as common conditions, and compares this interaction with their approach when faced with a familiar case that may not require support. Contributions include the following:

1. Exploration of KG-enhanced LLM in Rare Case Support: Investigates how clinicians selectively use AI assistance for rare or atypical cases, highlighting contexts where decision support is most valuable.
2. Understanding Clinician–AI Interaction Patterns: Observes clinician behaviour across familiar and unfamiliar cases to identify patterns of selective adoption, reliance, and impact on diagnostic confidence.

2. Related Works

LLMs have advanced natural language processing (NLP), enabling capabilities in text generation, summarisation, and semantic interpretation, supporting CDSS, NLP processing for electronic health records, medical question-answering systems, and healthcare education [4]. However, the reliability of LLM-generated content in medical contexts remains a significant concern, as limited exposure to curated medical data during training increases the risk of factual inaccuracies [14], deviations from established guidelines [15], and the amplification of biases, including those related to ethnicity [16]. Such limitations pose risks, particularly in high-stakes applications, as over-reliance on training data can lead to diagnostic errors, which may cause inappropriate treatment, unnecessary interventions, and significant harm [17,18].

To address these limitations, recent research has explored combining LLMs with KGs, leading to two primary integration paradigms that focus on LLM enhancements: KG-enhanced LLMs and synergised LLMs + KGs [19]. KG-enhanced LLMs incorporate structured knowledge to improve the accuracy, consistency, and interpretability of model outputs during different stages of the LLM cycle, while in a bidirectional, synergised LLM–KG integration, both systems iteratively support each other [20].

KGs support LLMs in handling complex queries by providing explicit relationships between entities, which helps LLMs reason over multiple connected concepts, resolve ambiguities, and generate outputs grounded in verified knowledge. This helps LLMs reduce hallucinations and enhance reasoning accuracy, which is particularly valuable in clinical settings where accuracy and traceability are essential [20]. They also enhance data integration, contextualisation, and decision-making, improving adaptability to real-world clinical scenarios [21]. However, these improvements remain constrained by the coverage and correctness of the underlying graph; any incompleteness or bias limits the precision of the resulting model [22].

https://doi.org/10.3390/electronics15030555

Many studies have built and evaluated the performance of LLMs when augmented with KGs, reporting promising improvements in reasoning, prediction, and classification tasks across different medical domains. Table 1 summarises recent representative works, highlighting their integration approach, medical application, methodology, models and datasets, evaluation metrics, and key results.

Table 1. Overview of representative studies integrating Knowledge Graphs with LLMs.

Paper	KG-LLM Integration Type	Medical Application	Evaluation Methods & Metrics	Key Outcomes
KG-Rank [15]	KG-enhanced LLM	Question & Answering system	Quantitative analysis of KG-based reranking framework using ROUGE-L, BERTScore, MoverScore & BLEURT	KG-based reranking improved QA performance; GPT-4 achieved +18% ROUGE-L improvement on ExpertQA-Bio dataset
DR. KNOWS [23]	Synergised Integration	Diagnostic Prediction	Quantitative analysis of KG path augmentation in prompt-based LLMs (ROUGE-2, ROUGE-L, CUI Precision, Recall, F1) & Human evaluation with 2 medical professionals	KG augmentation improved ChatGPT(GPT-3.5-Turbo) diagnostic F1 from 20.96 to 26.02 (5-shot). Human evaluation confirmed a 5% absolute improvement in correct diagnostic rationale agreement ($p < 0.001$).
XAI-Based CDSS [11]	Synergised Integration	Clinical Decision Support	Quantitative analysis of classification tasks using Precision, Recall, F1	KG integration led to modest F1 gains across mental health detection tasks, including a +0.1291 increase for Cause/Factor Detection with the Joint RoBERTa model.
MedIKAL [10]	Synergised Integration	Disease Diagnosis	Quantitative analysis of automatic diagnosis on datasets; comparison with LLMs, KG-enhanced LLMs, and synergised KG-LLMs using Precision, Recall, F1	Outperformed all other models including KG-Rank on CMD dataset on all metrics

While these studies demonstrate the potential of KG-enhanced LLMs, their evaluations largely emphasise correctness, factuality, or text similarity metrics such as ROUGE and BLEU. Although such metrics demonstrate measurable improvements on standard benchmarks, they provide limited insight into the practical utility of these models in supporting clinician reasoning. These metrics capture surface-level performance but fail to reflect critical aspects of clinical decision-making, including reasoning quality, reliability, and the ability to justify outputs in complex or uncertain scenarios. Consequently, a disconnect remains between prevailing evaluation frameworks and the real-world requirements of support systems, where nuanced reasoning and clinically meaningful guidance are essential.

Even when human evaluation is included (e.g., Gao et al., 2025 [23]), it often measures agreement with expert labels or retrospective performance on benchmark tests rather than

https://doi.org/10.3390/electronics15030555

on clinician behavior, selective adoption, or decision-making under real-world diagnostic uncertainty [24]. As a result, there is little understanding about how structured knowledge integration affects clinician interaction, reliance, and the likelihood of clinical adoption in real-world workflows. As a result, little is known about how structured knowledge integration affects clinician reliance, interaction patterns, or adoption in real-world workflows. Moreover, the heavy reliance of LLMs on effective prompting means that model use depends not only on model capabilities but also on clinician habits, experience with prompting, and expectations of the tool, all of which can influence perceived usefulness, efficiency, and willingness to adopt. This has practical implications: verbose outputs or repeated prompting may reduce efficiency and lead to selective adoption [25].

To address this gap, our study shifts the focus from model-centric performance evaluation to clinician-centred assessment. Rather than aiming to establish performance superiority, this work adopts an exploratory, formative approach to understand how clinicians interact with a KG-enhanced LLM across different diagnostic scenarios. We observe how KG-enhanced LLMs are used differently depending on diagnostic uncertainty, capturing interactions, trust, reasoning, and confidence, emphasising practical utility.

3. Materials and Methods

This section describes the design of a KG-enhanced LLM and an exploratory evaluation framework aimed at understanding how such systems may support clinical reasoning in rare disease diagnosis. Rather than optimising or benchmarking model performance, the focus of this study is on clinician interaction, perceived utility, and trust when engaging with a KG-enhanced diagnostic tool. The proposed framework embeds structured clinical context into LLM responses to promote grounded, interpretable outputs.

3.1. Pseudohypoparathyroidism: Disease and Case Selection

Pseudohypoparathyroidism (PHP) encompasses rare endocrine disorders characterised by end-organ resistance to parathyroid hormone (PTH), with subtypes including type 1A, type 1B, type 1C, pseudo-PHP, and type 2 [26]. Given the high level of clinical suspicion required to distinguish PHP from conditions such as idiopathic epilepsy or other causes of hypocalcaemia, an LLM augmented with a structured MKG could assist clinicians by systematically analysing symptom patterns and laboratory findings. Three case studies were selected for evaluation. The first two focus on PHP subtypes, and the third involves a common condition not included in the KG, serving as a control. This control ensures the KG does not provide information outside its scope and helps establish a baseline for clinician confidence when handling familiar conditions versus rare diseases. Quantitative and qualitative methods assess diagnostic accuracy, clinical relevance, and practical utility, offering insights into the potential of KG-enhanced LLMs to reduce misdiagnosis in complex cases.

3.1.1. Case Study 1 (Typical Presentation: PHP Type 1A)

Based on Najim et al. (2020), this scenario describes a 34-year-old woman who presented with symptomatic hypocalcaemia and was ultimately diagnosed with PHP type 1A [27]. Laboratory investigations revealed abnormal calcium, phosphate, and parathyroid hormone levels consistent with hormonal resistance. Additionally, the patient exhibited features of Albright hereditary osteodystrophy (AHO), consistent with the classical presentation of PHP type 1A, a rare but clinically important condition that is often underdiagnosed.

https://doi.org/10.3390/electronics15030555

3.1.2. Case Study 2 (Atypical Presentation: Pseudo-PHP)

Adapted from Najim et al. (2020), this case involves a 9-year-old girl attending a routine check-up to monitor growth given her short stature [27]. She had no clinical complaints, and laboratory findings were normal. Despite the absence of biochemical abnormalities, the child exhibited features characteristic of AHO, including a round face, short stature, and brachydactyly. The constellation of findings suggested pseudo-PHP, an atypical variant in which phenotypic features are present without hormonal resistance.

3.1.3. Case Study 3 (Control: Severe Malaria, Out-of-Scope Condition)

As a control, this case centres on a 55-year-old woman who developed severe Plasmodium falciparum malaria following a trip to Ghana [28]. Upon returning to Florida, she was admitted with fever, confusion, and hypotension and was treated successfully with intravenous artesunate. Because malaria falls outside the KG's scope, this case was included to assess hallucination risk when the system encounters conditions without KG coverage.

3.2. KG Construction

The KG was manually constructed from scientific publications, including the peer-reviewed literature, textbooks, and clinical guidelines issued by authoritative institutions, publishers, and researchers. These sources are highly trustworthy, widely available, and provide a reliable foundation for creating a disease-specific KG. Manual construction was chosen to ensure careful curation of clinically relevant entities and relationships, minimising errors or omissions that automated methods might introduce. The focus was on PHP and one of its common misdiagnoses: epilepsy [29]. This diagnostic error can occur when patients present with seizure-like complications caused by chronic hypocalcaemia or when tetany is mistaken for seizures. This overlap highlights the need for careful curation of clinically relevant knowledge.

We manually extracted relevant entities and relationships from these sources. Extraction focused on key clinical features, diagnostic criteria, treatment options, and ways in which PHP is commonly confused with epilepsy. Entities and connections were organised into structured sets to identify critical relationships and construct a comprehensive knowledge base. A simple, well-defined graph schema was designed in Neo4j (Desktop 5.x) to capture both hierarchical and semantic relationships. Hierarchical relationships represent subtype structures, such as the IS_A link between PHP and its subtypes PHP type 1A and type 1B. Semantic relationships reflect clinical associations across entity types; for example, epilepsy has a HAS_SYMPTOM relationship with seizure and a DIAGNOSED_BY link to EEG (Figure 1). This schema supports meaningful clinical queries and enables complex reasoning across diagnostic pathways and differential diagnoses. Each node was populated with a detailed description of the associated entity. Nodes were also assigned properties to encapsulate relevant clinical information and salient characteristics. This enabled accurate, enriched representations of entities, making it easier to trace relationships and identify potential diagnostic patterns.

In total, the KG comprises 160 nodes, 252 edges, 69 node labels, and 71 relationship types. The KG's scope is small but focused, centering on PHP and selected related details, such as overlapping features with epilepsy, for example, differentiating seizures from tetany. While the size is limited, this was intentional for an exploratory study, allowing for the careful assessment of clinician interaction and system utility.

https://doi.org/10.3390/electronics15030555

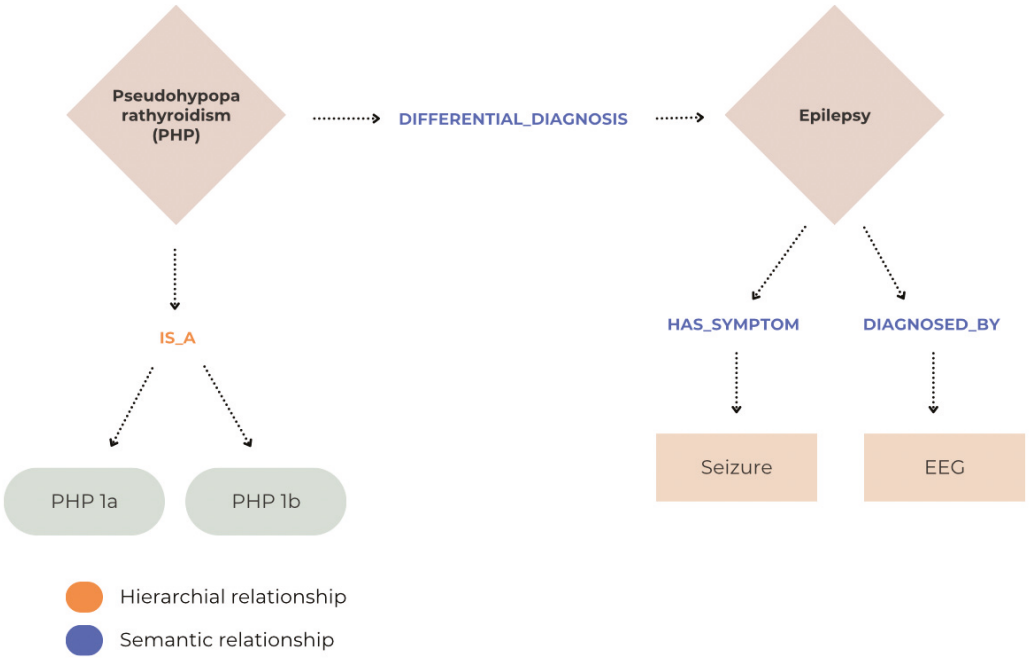


Figure 1. This schematic illustrates examples of hierarchical and semantic relationships in the Knowledge Graph. Hierarchical links represent subtype structures (e.g., PHP → PHP type 1A), while semantic links capture clinical associations (e.g., epilepsy HAS_SYMPTOM seizure).

3.3. LLM Integration

To support the KG-enhanced system, GPT-4o-mini (version date: 18 July 2024) was selected for its predictable reasoning and computational efficiency [30]. Compared with GPT-3.5-turbo, GPT-4o-mini demonstrates approximately four times the reasoning capacity and operates at roughly three times the processing speed, while supporting multimodal inputs and extended context lengths. It was chosen over other medical LLMs not for maximal diagnostic accuracy, but to enable smooth, low-latency interactions that allow clinicians to explore and evaluate the system's utility without introducing variability or unnecessary complexity.

The integration follows a straightforward, context-enriched framework. When a user query is received, relevant triples are retrieved directly from the Neo4j knowledge graph using Cypher queries. These triples are structured into a textual context and injected into a fixed prompt for GPT-4o-mini, which generates responses constrained to the KG content. This ensures that the LLM's reasoning is grounded in structured medical knowledge, avoiding hallucinations while providing context-aware guidance.

LangChain orchestrates the workflow, combining query handling, KG retrieval, and prompt construction into a seamless pipeline.

The fixed prompt template used to constrain the LLM's responses to the retrieved knowledge graph context is provided in Appendix A.

3.4. System Architecture

The KG-enhanced LLM system uses a lightweight, modular design tailored for exploratory evaluation. The backend is implemented in Python 3.12 with FastAPI, exposing a single synchronous API endpoint. Queries are processed sequentially, with responses generated only after full KG retrieval and LLM reasoning.

The KG is stored in Neo4j (Desktop 5.x), and a custom service layer retrieves subject–predicate–object triples, including node labels, descriptions, and properties. Retrieved triples are structured and serialized into a textual context for the LLM. No ranking, thresholds, similarity filtering, or re-ranking mechanisms are applied.

https://doi.org/10.3390/electronics15030555

GPT-4o-mini handles reasoning via LangChain in a zero-shot configuration, using a fixed prompt that instructs the model to respond solely with the KG context and indicate when information is unavailable. Default model parameters are used, including a temperature of 0.7. Response times reflect the cumulative cost of KG retrieval, context preparation, and LLM generation; no latency measurements or optimization strategies were applied (see Table 2).

The frontend is minimal, featuring a query input box, system title, and response display area. This simple interface ensures that clinician focus remains on the system's reasoning support rather than the interaction design.

Table 2. Key Parameters and Design Choices of the KG-Enhanced LLM Framework.

Parameter	Setting/Description
Knowledge Graph Retrieval	
Knowledge retrieval strategy	Direct graph traversal over Neo4j using Cypher queries
Retrieval limit (Top-*k*)	Not explicitly parameterised; up to 1000 triples retrieved per query
Similarity metric	Not applicable (no embedding-based similarity search)
Re-ranking	Not enabled
Filtering thresholds	None applied
Context Handling	
Context formatting	Subject–relation–object triples rendered as text
Context injection method	Full retrieved context appended to the LLM prompt
LLM Configuration	
LLM model	GPT-4o-mini (via LangChain)
LLM temperature	0.7
System Behavior/Constraints	
Abstention behaviour	Implicit, enforced via prompt instruction when KG coverage is insufficient
Response grounding	Restricted to provided KG context via prompt constraint
Latency optimisation	Not explicitly optimised or measured

3.5. Evaluation Process

The evaluation adopts a formative, mixed-methods design intended to explore clinician interaction with a KG-enhanced LLM rather than to establish definitive performance gains. Evaluation is divided into two stages: (1) a limited technical assessment to ensure basic system reliability and faithfulness, and (2) a clinician-centered evaluation focused on usability, trust, and perceived support for diagnostic reasoning (Figure 2).

3.5.1. Model Evaluation

Model evaluation follows a structured and automated approach using the Retrieval-Augmented Generation Assessment System (RAGAS). A dataset of 10 clinically relevant questions with expert-validated reference answers was prepared, balancing short-form and long-form queries. The RAGAS evaluation framework, designed for RAG systems [31], comprises five components: faithfulness, context precision, context recall, answer relevancy, and answer correctness. These metrics are computed automatically by the framework.

Faithfulness reflects factual accuracy, calculated as the number of correct facts divided by the total number of facts in the response, ensuring the system avoids introducing misleading or incorrect information. Answer relevance measures the proportion of relevant concepts in a response, indicating whether outputs address the clinical query meaningfully. Context precision captures the proportion of retrieved sentences that are relevant, highlighting retrieval efficiency, while context recall evaluates whether the system retrieves all relevant KG knowledge available for the query. Answer correctness combines semantic similarity and factual accuracy to assess alignment with validated ground truth.

https://doi.org/10.3390/electronics15030555

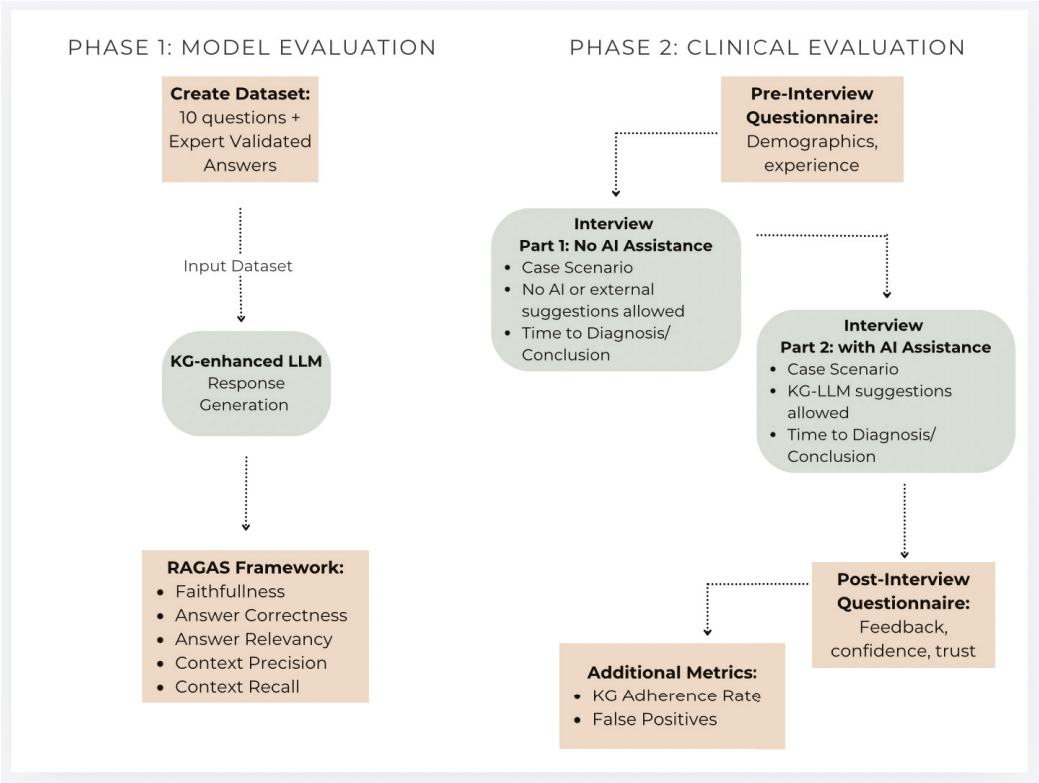


Figure 2. Overview of the evaluation process. The model evaluation uses a dataset of expert-validated questions assessed with RAGAS to measure faithfulness, relevance, and KG retrieval performance. The clinical evaluation involves a two-phase simulation where clinicians first diagnose cases unaided, then with KG-enhanced LLM support, capturing diagnostic accuracy, confidence, and adherence to AI recommendations.

While ROUGE metrics (Recall-Oriented Understudy for Gisting Evaluation) were also considered, their applicability was limited by the small sample size and the exploratory nature of this study. The evaluation prioritised contextual relevance, transparency, and faithfulness over surface-level text-overlap metrics such as ROUGE.

3.5.2. Clinical Evaluation

Clinical evaluation examines how clinicians interact with the KG-enhanced LLM and perceive its utility in diagnostic decision-making. The evaluation consists of a pre-interview survey, a two-phase clinical simulation, and a post-interview survey to capture both performance and perceived clinical utility. Before the simulation, participants complete a pre-interview survey to collect demographic data, including years of experience and specialty. During the interview, participants are presented with three test cases: two corresponding to diseases represented in the KG (Cases 1 and 2, differing in subtype and complexity) and one control case featuring a disease outside the KG (Case 3). This design allows for the assessment of the system's ability to manage both familiar and "out-of-scope" scenarios, including appropriate handling of uncertainty.

The simulation occurs in two phases. In Phase 1, participants complete the cases without AI assistance. They record their diagnostic conclusions and time-to-diagnosis while the interviewer acts as the patient, responding to history-taking questions. Participants may request physical examination findings, laboratory tests, and radiology results, which are provided according to the case and clinical judgment. The order of case presentation varied: the first participant completed Case 2 → Case 1 → Case 3; the next eight participants followed Case 1 → Case 2 → Case 3, starting with a straightforward case before progressing to a more atypical one; and the final participant, with more endocrinology experience,

https://doi.org/10.3390/electronics15030555

completed Case 3 → Case 1 → Case 2 to explore the effect of starting with the most familiar condition.

In Phase 2, participants complete the same cases with access to the KG-enhanced LLM. They again record diagnostic conclusions, adherence to KG suggestions, and time-to-diagnosis. Time-to-diagnosis is treated as a secondary, descriptive outcome. Because participants have prior exposure to the cases and interactions with the AI include typing and prompting, these results do not allow for causal inference and are reported for illustrative purposes only.

The primary evaluation focuses on three key clinical outcomes: Diagnostic Assessment, confidence, and KG adherence. Diagnostic accuracy measures whether the participant reaches the correct diagnosis, including subtypes when applicable. KG adherence reflects the proportion of AI recommendations integrated into the final diagnosis, classified as full (AI suggestions fully incorporated and leading to the correct diagnosis), partial (some engagement with AI outputs but the case is not fully resolved), or none (AI disregarded or no diagnosis reached). Diagnostic confidence is measured using a 5-point Likert scale before and after AI interaction, recorded during post-interview feedback, to capture changes attributable to KG assistance. Post-interview feedback also captures participant perceptions of accuracy, relevance, usability, trust, and overall satisfaction with the AI-assisted workflow. Secondary metrics include time-to-diagnosis and observed instances where the model provided misleading or incorrect suggestions outside its scope. These instances were recorded descriptively, with a target threshold of <10%, but formal statistical false-positive rates were not calculated. Figure 3 shows the user interface used by clinicians to submit queries and view KG-enhanced LLM responses during the evaluation.

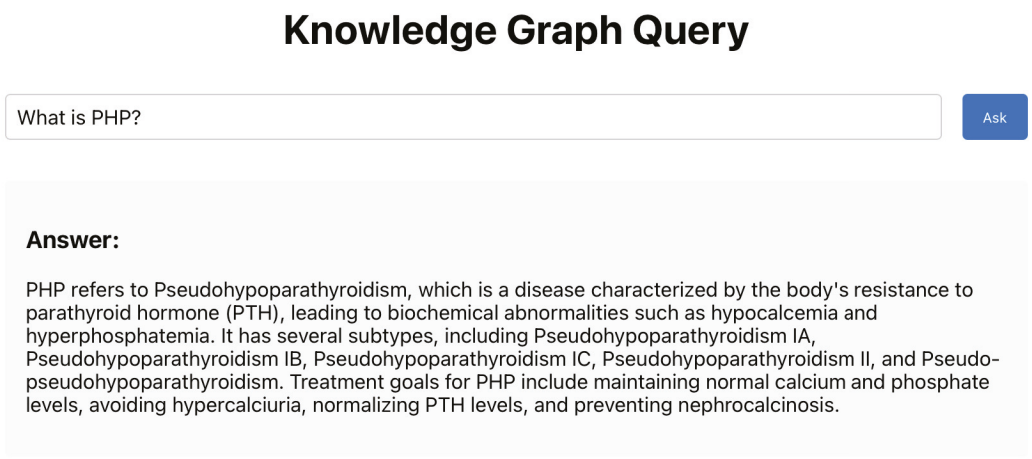


Figure 3. User interface.

3.5.3. Participant Selection Rationale

For this pilot, the target group consists of 10–15 clinicians, including general practitioners (GP), residents, and junior doctors with 1–7 years of experience. This mix allows for diverse perspectives, ensuring the KG-enhanced LLM is evaluated by those most likely to benefit from decision support. Novice clinicians (1–3 years) are particularly likely to improve efficiency and confidence with KG assistance, as they tend to rely more on external support than experienced clinicians [32]. They may also exhibit the greatest gains in diagnostic confidence. Mid-level clinicians (4–7 years) provide valuable insight into KG usefulness for atypical cases, where clinical experience may be limited. Focusing on this group also avoids potential bias from expert clinicians (10+ years), who may dismiss the KG due to overconfidence in their diagnostic skills. This participant profile aligns with assessing the feasibility and early utility of the KG in real-world settings. This selection

supports the study's aim of generating early, clinician-informed insights to guide future system design and evaluation, rather than establishing generalisable effectiveness claims.

3.6. Ethical Considerations

Ethical principles were paramount, and we ensured compliance with key guidelines and regulations governing AI in healthcare. These steps ensured adherence to clinical ethical standards while addressing concerns related to generative AI in diagnostic decision-making.

3.6.1. Patient Data Privacy and Confidentiality

One critical ethical consideration was patient data privacy. To avoid risks related to sensitive data, participants were medical professionals, and cases were based entirely on published or edited materials with no personal information. This adheres to the General Data Protection Regulation (GDPR), which mandates explicit consent and transparency in the use of patient data [33]. Using simulated cases avoided real clinical settings and ensured no patient data were compromised. This process aligned with GDPR principles of protecting personal data and securing patient confidentiality.

3.6.2. Clinical Oversight and AI Limitations

The National Institute for Health and Care Excellence (NICE) guidelines stress the importance of healthcare professionals reviewing AI outputs before making clinical decisions [34]. We followed this principle by ensuring that the AI-driven, KG-enhanced LLM outputs were not solely relied upon for final diagnosis but used only to assist clinicians. A medical professional reviewed all diagnostic decisions based on AI recommendations, mitigating risks associated with over-reliance on AI. Moreover, NICE guidelines highlight that AI systems use fixed algorithms in clinical settings, limiting their ability to adapt to real-time data [35]. By conducting the study in a controlled environment where clinicians retained authority over final diagnoses, we ensured that AI complemented, rather than replaced, clinical judgement.

3.6.3. Risk Assessment and Transparency

In line with the G7 AI Code of Conduct, which advocates continuous risk assessment and transparency, the study prioritised transparency in its methodologies and results [36,37]. Detailed information about AI capabilities, limitations, and data used was made available to all participants. This ensured clinicians were well-informed about system operation, strengths, and constraints.

3.6.4. Cultural Sensitivity and Inclusivity

Ethical guidelines also call for consideration of cultural factors in healthcare. Although cases varied in age, gender was not considered a differentiating factor, in line with real-world clinical cases. This choice reflected the need to represent a broad spectrum of patient demographics. Recognising that cultural factors can influence diagnosis and care, future studies should incorporate a broader range of cultural contexts to align with evolving standards for inclusivity and cultural sensitivity.

3.6.5. Medical Professional Involvement

Throughout the study, medical professionals played integral roles in development and evaluation phases, addressing concerns about over-reliance or potential misuse of technology in clinical practice. Active clinician involvement ensured appropriate oversight of AI use. Consistent with NICE and the G7 AI Code of Conduct, healthcare professionals retained control over AI-generated findings, with AI as a supportive tool rather than a

https://doi.org/10.3390/electronics15030555

decision-making authority. This approach aligns with ethical principles that prioritise human expertise in healthcare. By ensuring AI enhanced, rather than replaced, clinical decision-making, the study emphasised the central role of clinicians in the diagnosis. This involvement also helped mitigate risks associated with over-reliance on AI systems, ensuring that final diagnoses remained with experienced medical professionals.

By following these ethical principles and complying with established guidelines, the study ensured responsible deployment of the AI model. These measures safeguarded patient privacy, maintained clinical oversight, and fostered clinicians' trust in AI technologies. Ultimately, by adhering to these ethical standards, the study aimed to establish a framework for responsible, transparent use of AI in healthcare to enhance diagnostic accuracy and support clinicians' decision-making.

4. Results

4.1. Model Evaluation Results

The primary aim of the model evaluation was to verify that a minimally viable, clinically coherent knowledge graph could be successfully queried and used by participants, rather than to benchmark model performance or establish generalisable accuracy claims. This section presents an exploratory evaluation of the KG-enhanced LLM using a small, curated question set (n = 10) with expert-validated answers (Table 3). The questions primarily address PHP and related endocrine features, with one question drawn from epilepsy. They span both fact-based knowledge (e.g., hormone function, genetic mutations, clinical features) and reasoning-oriented tasks (e.g., interpreting subtype characteristics and management considerations). Each question was submitted directly to the KG-enhanced LLM without additional context, instructions, or prompt engineering.

Table 3. Dataset of clinical questions with expert-validated answers and corresponding KG-enhanced LLM responses.

ID	Question	Expert Validated Answer	KG-Enhanced LLM Response
1	What is the primary function of PTH?	Regulation of calcium and phosphate concentrations	The primary function of PTH is to increase calcium levels in the blood.
2	Vitamin D deficiency is associated with which type of PHP?	PHP type 2	Vitamin D deficiency is associated with PHP type 2.
3	Which PHP subtypes have a preserved cAMP response?	Pseudo-PHP and PHP type 2	The PHP subtypes that have a preserved cAMP response are Pseudo-PHP and PHP type 2.
4	Which genetic mutation is associated with PHP?	GNAS mutation	The genetic mutation associated with PHP is the GNAS gene mutation.
5	AHO is typically associated with which types of PHP?	PHP types 1a, 1c, Pseudo-PHP, and sometimes type 2	AHO is typically associated with the following types of PHP: PHP type 1a, Pseudo-PHP, PHP type 1b, PHP type 1c, and PHP type 2.
6	What clinical features are characteristic of AHO?	Short stature, obesity, round face, brachydactyly, subcutaneous ossifications	The clinical features characteristic of AHO include: short stature, round facies, brachydactyly, obesity, and osteoma cutis.
7	How is Chvostek's sign elicited on physical exam?	Tap the cheek about 2 cm in front of the ear, over the facial nerve	Chvostek's sign is elicited on physical exam by performing facial nerve percussion.

https://doi.org/10.3390/electronics15030555

Table 3. *Cont.*

ID	Question	Expert Validated Answer	KG-Enhanced LLM Response
8	When managing PHP, what should be prevented or avoided?	Hypercalciuria	When managing PHP, the following should be prevented or avoided: hypercalciuria and severe symptomatic hypocalcemia.
9	Which parameters should be monitored during long-term management of PHP?	Serum calcium, phosphate, PTH, renal function, urinary calcium	During long-term management of PHP, the following parameters should be monitored: serum PTH, serum calcium, serum phosphate, and 24 h urine calcium. Additionally, PHP treatment monitoring is recommended.
10	Which age groups are associated with febrile seizures?	Infants and children	Febrile seizures are seen in infants and children.

The curated questions and corresponding model responses were evaluated using the RAGAS framework, and the resulting scores (Table 4) confirm functional viability and surface obvious failure modes of the KG–LLM integration, rather than providing definitive measures of model performance or robustness.

Table 4. RAGAS Metric Results.

Question	Faithfulness	Context Precision	Context Recall	Answer Relevancy	Answer Correctness
Q1	1.00	1.00	1.00	1.00	1.00
Q2	1.00	0.70	1.00	0.88	1.00
Q3	1.00	0.92	1.00	0.99	0.85
Q4	0.00	0.20	0.00	1.00	1.00
Q5	0.67	0.33	0.80	0.89	1.00
Q6	0.60	1.00	0.67	0.97	1.00
Q7	0.50	1.00	1.00	0.99	1.00
Q8	0.67	1.00	1.00	0.99	1.00
Q9	0.67	0.70	0.60	1.00	1.00
Q10	1.00	0.75	1.00	0.97	1.00
Average	0.71	0.76	0.81	0.97	0.98

Across the dataset, answer relevancy (mean = 0.97) and answer correctness (mean = 0.98) were consistently high, indicating strong alignment between model outputs, clinical questions, and expert-validated answers. Context recall (0.81) was also relatively strong, suggesting that the system generally retrieved relevant entities from the KG. Although context precision (0.76) and faithfulness (0.71) exhibited greater variability, the overall mean RAGAS score (0.85) reflects acceptable retrieval and response alignment at the aggregate level.

To further characterise retrieval balance, an F1 score was calculated using context precision and recall:

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}.$$

Based on an average precision of 0.76 and recall of 0.81, the resulting F1 score was

$$\mathrm{F1} = 2 \times \frac{0.76 \times 0.81}{0.76 + 0.81} \approx 0.79.$$

This F1 score indicates a reasonably balanced level of retrieval accuracy and completeness.

However, aggregate performance masks important failure modes that become apparent at the question level (Table 4). Lower context precision indicates that, while most retrieved information was relevant, irrelevant details were occasionally included. Such "noise" can increase cognitive load, requiring clinicians to filter extraneous information to

https://doi.org/10.3390/electronics15030555

extract clinically useful insights. Similarly, reduced faithfulness increases the risk of hallucinations, which can undermine clinical safety. This divergence highlights a fundamental limitation of automated RAG metrics: clinically correct answers may still be insufficiently grounded in retrieved evidence, while minor abstraction or paraphrasing may be penalised as faithfulness errors despite remaining clinically valid.

Several question-level examples illustrate these limitations. In Q4, the model achieved perfect answer correctness (1.00) despite scoring 0.00 for both faithfulness and context recall and only 0.20 for context precision. Although the response was factually correct, it was entirely ungrounded in the retrieved knowledge graph, representing a breakdown of the intended safeguard against unsupported answers. In Q5, reduced faithfulness (0.67) and low context precision (0.33) reflected partial hallucination, where correct information was combined with errors, including inappropriate inclusion of PHP type 1B and omission of subtype occurrence details. In contrast, Q7 demonstrated a different failure pattern: faithfulness declined to 0.50 despite strong performance across other metrics because a clinically acceptable paraphrase ("facial nerve percussion") omitted specific procedural detail ("tap 2 cm anterior to the ear"). While less concerning than outright hallucination, this example highlights the sensitivity of faithfulness metrics to phrasing.

Overall, average performance indicates that the KG-enhanced LLM is sufficiently accurate and relevant for participants to meaningfully engage, although sensitivity to phrasing and occasional retrieval noise may pose minor challenges.

4.2. Clinical Evaluation Results

4.2.1. Pre-Interview Survey

Ten participants were included in the study. The information gathered from the pre-interview survey, including general demographic characteristics, is summarised in Table 5. Additional items assessed in the pre-interview survey included participants' use of search or AI tools, perceived trust in these tools, perceived helpfulness of responses, and frequency of use, as summarised in Table 6. Furthermore, the types of tools used are shown in Figure 4, along with the primary purposes for which they were applied in Figure 5.

Table 5. Demographic Information of Participants. GOPD refers to General Outpatient Department. GP refers to General Practitioner. N/A refers to Not Applicable, used when participants are not currently working.

ID	Age	Practice	Department	Role	Years of Experience
DR01	20–29	Private	Internal Medicine	GP	1–3 years
DR02	30–39	Private	GOPD	GP	1–3 years
DR03	20–29	N/A	–	GP	1–3 years
DR04	20–29	Private	Paediatrics	GP	1–3 years
DR05	20–29	Private	GOPD	GP	1–3 years
DR06	20–29	Government	Emergency	GP	1–3 years
DR07	20–29	Private	Paediatrics	GP	1–3 years
DR08	20–29	N/A	–	GP	1–3 years
DR09	30–39	Government	GOPD	GP	3–5 years
DR10	20–29	Government	Internal Medicine	Junior Resident	1–3 years

Participants completed a survey to capture their prior experience with AI and search tools (Table 6). Responses were collected using structured scales:

- Use of Tools: Whether participants had used AI or search tools before (Yes/No).
- Helpfulness: How useful they perceived the tools to be (Not Helpful, Neutral, Helpful, Very Helpful).
- Trust: Level of trust in the tool's outputs (Never, Rarely, Somewhat, Mostly, Completely).

https://doi.org/10.3390/electronics15030555

- Frequency of Use: How often they used the tools (Never, Rarely, Monthly, Weekly, Daily).

Table 6. Pre-Interview Survey Responses on AI and Search Tool Usage.

ID	Use of Tools	Helpfulness	Trust	Frequency of Use
DR01	Yes	Neutral	Somewhat	Weekly
DR02	Yes	Helpful	Somewhat	Weekly
DR03	Yes	Very Helpful	Mostly	Weekly
DR04	Yes	Helpful	Somewhat	Rarely
DR05	Yes	Very Helpful	Mostly	Weekly
DR06	Yes	Very Helpful	Mostly	Rarely
DR07	Yes	Helpful	Mostly	Daily
DR08	No	–	–	–
DR09	Yes	Helpful	Mostly	Weekly
DR10	Yes	Very Helpful	Mostly	Daily

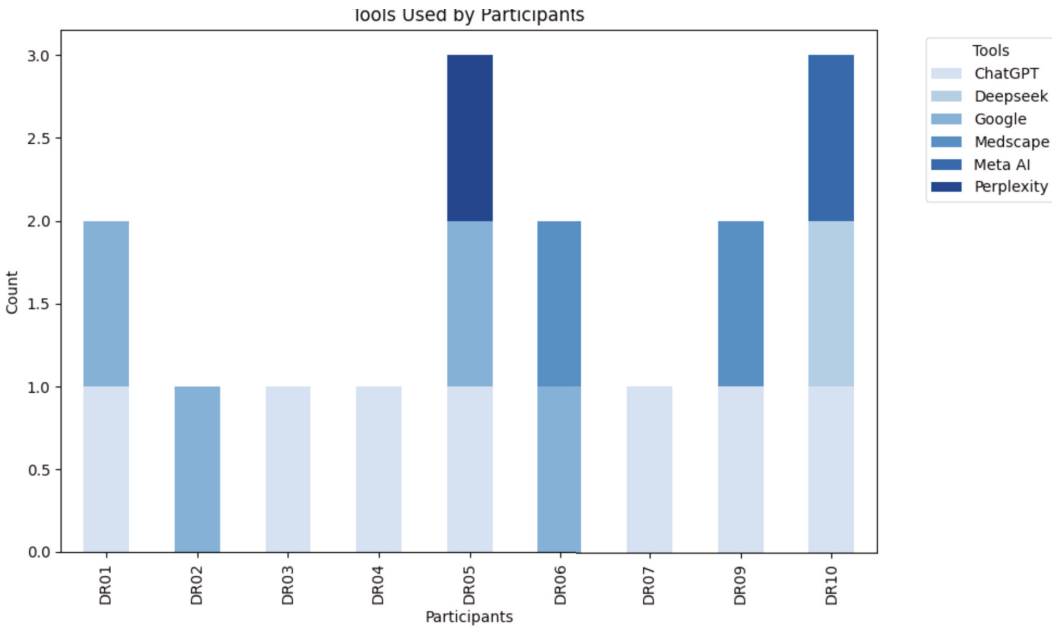


Figure 4. AI and search tools used by individual participants. Each bar represents a participant (n = 10), showing which tools they reported using, including ChatGPT (7 participants), Google (4), Medscape (2), Perplexity (1), and others.

4.2.2. Diagnostic Assessment & Adherence

Diagnostic performance was assessed descriptively, focusing on whether participants identified the relevant condition or included appropriate differentials, including subtypes where applicable. Participants were not required to reach a definitive diagnosis; instead, they reported their differential diagnoses and conclusions when faced with uncertainty. These actionable conclusions, recorded in Tables 7 and 8, provide insights into how the KG-enhanced LLM may support clinical reasoning.

Adherence patterns were analysed for Cases 1 and 2, in which all ten participants engaged with the KG-enhanced LLM. Case 3 was excluded, as only three participants used AI in this case; the remaining eight opted not to engage with the system, citing high diagnostic confidence and satisfaction with their own clinical reasoning. Across Cases 1 and 2, this resulted in 20 instances in which participants attempted to reach a diagnosis using the KG-enhanced LLM (Table 9).

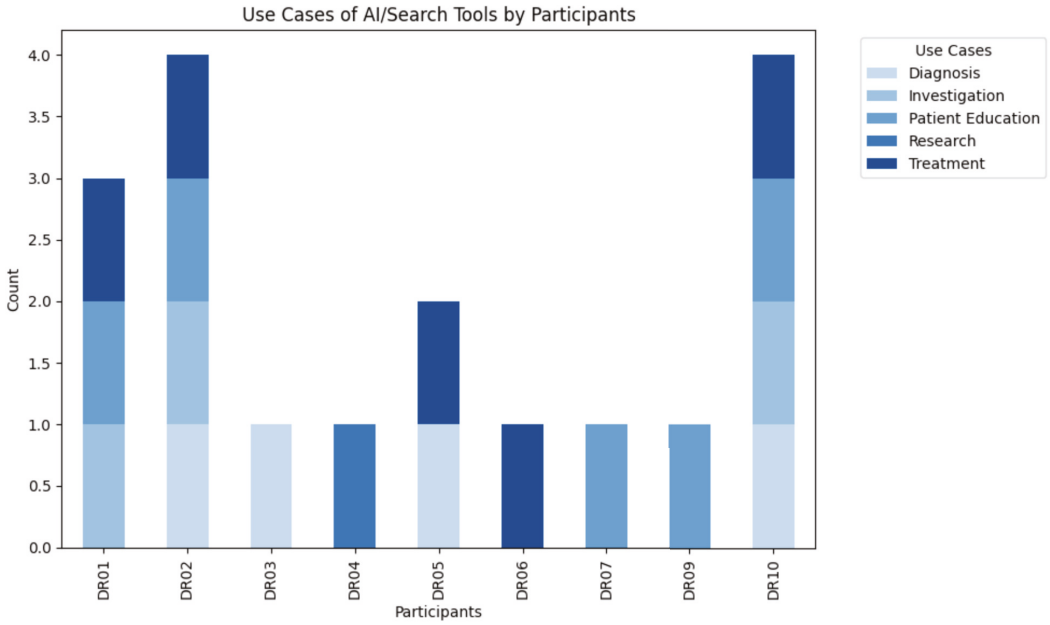


Figure 5. Use cases of AI and search tools reported by individual participants. Each bar represents a participant (n = 10), showing the clinical purposes for which they reported using AI or search tools, including diagnosis (4 participants), Patient education (4 participants) and others.

Table 7. Case 1: Diagnostic Assessment and KG Adherence (PHP Type 1A).

ID	No AI Conclusion	AI Conclusion	Adherence
DR01	Manage Symptoms	PHP, no subtype	Partial
DR02	Referral	PHP type 1A	Full
DR03	Referral	PHP, no subtype	Partial
DR04	Secondary Hypoparathyroidism	PHP, no subtype	Partial
DR05	Hypocalcaemia	Inconclusive	No
DR06	Referral	PHP, no subtype	Partial
DR07	Inconclusive	PHP, no subtype	Partial
DR08	Hypoparathyroidism	PHP, no subtype	Partial
DR09	Referral	PHP, no subtype	Partial
DR10	PHP, no subtype	PHP, no subtype	Partial

Table 8. Case 2: Diagnostic Assessment and KG Adherence (Pseudo-PHP).

ID	No AI Conclusion	AI Conclusion	Adherence
DR01	Inconclusive	Inconclusive	No
DR02	Referral	Pseudo-PHP	Full
DR03	Referral	PHP, no subtype	Partial
DR04	Cushing's Syndrome	Pseudo-PHP	Full
DR05	Referral	Inconclusive	No
DR06	Referral	PHP, no subtype	Partial
DR07	Referral	Pseudo-PHP	Full
DR08	Referral	Pseudo-PHP	Full
DR09	Referral	Pseudo-PHP	Full
DR10	Referral	PHP, no subtype	Partial

In three out of twenty instances (15%), the KG-enhanced LLM was unable to provide relevant or usable diagnostic support. In each of these cases, the system explicitly communicated its limitation with messages such as "This information is not contained in my knowledge base." A similar outcome occurred when three participants prompted the KG-enhanced LLM in the control case. Importantly, in none of the twenty instances did the KG-enhanced LLM provide misleading, incorrect, or out-of-scope information.

Table 9. Summary of Participant adherence to KG-enhanced LLM suggestions.

Case	Full Adherence	Partial Adherence	No Adherence	Total Instances	Adherence Rate
Case 1	5	3	2	10	80%
Case 2	1	8	1	10	90%
Total	6	11	3	20	85%

4.2.3. Time/Efficiency Analysis

To examine whether the KG-enhanced LLM influenced diagnostic efficiency during the interview, time-to-diagnosis was analysed as a secondary outcome. Average and median completion times were calculated for Cases 1 and 2. Case 3 was excluded from this analysis, as it served as a control condition and involved minimal AI usage. Time differences were computed at the participant level by subtracting AI-assisted completion times from non-AI times (Table 10), where positive values indicate faster task completion with AI assistance.

Across nine participants, excluding missing data from DR05, the average time difference was 135.5 s, with a median of 141 s and a standard deviation (SD) of 81.5 s (Table 11). These values indicate that, on average, participants took longer to complete tasks without AI, suggesting that the use of AI improved efficiency in this context.

A paired-samples t-test was conducted to assess whether the observed improvement in time efficiency with AI was statistically significant. This test was appropriate because each participant completed tasks under both AI and non-AI conditions, allowing for direct within-subject comparisons. For each participant, the average completion times for Case 1 and Case 2 were calculated separately for the AI and non-AI conditions. The test statistic was calculated using the following formula:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

where $\bar{d}$ is the mean of the paired differences, s_d is the standard deviation of these differences, and n is the number of valid paired observations. The degrees of freedom for the test were calculated as the number of valid paired comparisons minus one:

$$df = n - 1 = 9 - 1 = 8,$$

reflecting the nine participants with complete paired data (excluding the missing AI value for DR05). The mean difference was $\bar{d} = 135.5$ s with a standard deviation of $s_d = 81.5$ s. This resulted in a test statistic of

$$t(8) = \frac{135.5}{81.5/\sqrt{9}} \approx 4.99,$$

with a corresponding probability under the null hypothesis of $p = 0.001$, indicating that the observed reduction in completion time with AI was very unlikely to have occurred by chance.

The effect size was calculated using Cohen's d_z for paired samples:

$$d_z = \frac{\bar{d}}{s_d} = \frac{135.5}{81.5} \approx 1.66,$$

indicating a very large effect of AI on time efficiency. A 95% confidence interval (CI) for the mean difference was computed as

$$\text{CI}_{95\%} = \bar{d} \pm t_{0.025,8} \cdot \frac{s_d}{\sqrt{n}} = 135.5 \pm 2.306 \cdot 27.17 \approx [73.8, 197.2].$$

https://doi.org/10.3390/electronics15030555

To check the robustness of these findings, we conducted a Wilcoxon signed-rank test including all valid paired observations (excluding the missing value for DR05). The Wilcoxon test statistic was calculated as:

$$W = 0.0, \quad p = 0.0039,$$

confirming that the reduction in completion time with AI was also statistically significant under this non-parametric sensitivity check.

Table 10. Participant-level Results (No AI vs. AI).

ID	Case 1 No AI	Case 1 AI	Case 2 No AI	Case 2 AI	No AI Avg	AI Avg	Difference
DR01	470 s	354 s	480 s	478 s	475.0 s	416.0 s	59.0 s
DR02	600 s	507 s	900 s	502 s	750.0 s	504.5 s	245.5 s
DR03	300 s	130 s	392 s	557 s	346.0 s	343.5 s	2.5 s
DR04	375 s	215 s	552 s	290 s	463.5 s	252.5 s	211.0 s
DR05	214 s	–	355 s	–	284.5 s	–	–
DR06	206 s	244 s	476 s	218 s	341.0 s	231.0 s	110.0 s
DR07	80 s	180 s	620 s	180 s	350.0 s	180.0 s	170.0 s
DR08	80 s	97 s	487 s	330 s	283.5 s	213.5 s	70.0 s
DR09	278 s	269 s	568 s	295 s	423.0 s	282.0 s	141.0 s
DR10	180 s	225 s	572 s	106 s	376.0 s	165.5 s	210.5 s

Table 11. Summary Statistics Across Participants excluding missing data (DR05).

Statistic	Case 1 No AI	Case 1 AI	Case 2 No AI	Case 2 AI	Difference
Average	285.4 s	246.8 s	560.8 s	328.4 s	135.5 s
Median	278.0 s	225.0 s	552.0 s	295.0 s	141.0 s
SD	174.3 s	123.4 s	144.3 s	154.5 s	81.5 s

4.3. Post-Interview Results

The post-interview survey collected changes in confidence and participants' evaluations of the AI-assisted tool.

4.3.1. Diagnostic Confidence

Changes in diagnostic confidence before and after AI use were analysed to assess whether the KG-enhanced LLM provided meaningful support during clinical reasoning. Confidence was captured using a five-point likert-type scale and converted into numeric change (Table 12) scores to enable comparison across cases and participants (Table 13).

Case 1: Most participants began with moderate confidence, with scores around "Neutral" (3 on the Likert scale). After using the KG-enhanced LLM, 7 out of 10 participants (70%) reported an increase in confidence, 2 participants (20%) experienced no change, and 1 participant (10%) reported a slight decrease. This indicates that the AI tool generally supported clinicians' confidence when dealing with this case.

Case 2: Participants initially showed mixed confidence, ranging from "Low" to "High" (2–4). Following interaction with the KG-enhanced LLM, 6 participants (60%) reported an increase in confidence, 3 participants (30%) had no change, and 1 participant (10%) experienced a decrease. The tool appeared to provide moderate benefit, particularly for those who initially had lower confidence.

Case 3: Most participants were already confident before AI assistance, with initial ratings mostly "High" or "Very High" (4–5). Eight participants (80%) who opted not to engage with the KG-enhanced LLM reported no change in confidence, while 2 participants (20%), despite receiving no helpful information from the AI, experienced a slight increase as they felt validated in their own knowledge.

https://doi.org/10.3390/electronics15030555

Table 12. Mapping of qualitative Likert-scale metrics to numeric values before and after KG-enhanced LLM interaction, allowing for easier interpretation of changes.

Qualitative Metric	Numeric Value	Qualitative Change	Numeric Value
Very Low	1	Decreased Significantly	−2
Low	2	Decreased Slightly	−1
Neutral	3	No Change	0
High	4	Increased Slightly	+1
Very High	5	Increased Significantly	+2

Table 13. Participant-level change across cases before and after KG-enhanced LLM interaction.

ID	Case 1 Before	Case 1 After	Case 2 Before	Case 2 After	Case 3 Before	Case 3 After
DR01	3	+1	3	+1	3	0
DR02	2	+2	2	+2	4	0
DR03	2	+2	2	+2	5	0
DR04	4	+1	4	+2	3	0
DR05	4	-2	4	-2	4	0
DR06	4	+2	4	+2	4	0
DR07	2	+1	2	+1	2	0
DR08	3	+1	3	+1	3	0
DR09	3	+1	3	+1	5	0
DR10	4	+1	2	+1	4	+1

4.3.2. Participant Feedback

Participant feedback was collected using 5-point Likert scales for five aspects of the KG-enhanced LLM (Table 14). The scales were defined as follows:

- Usability: Overall ease of using the KG-enhanced LLM, rated on a 5-point scale from Very difficult to Very easy (Very difficult, Difficult, Neutral, Easy, Very easy).
- Relevance of Responses: How often the AI responses provided relevant and useful information to aid clinical diagnosis, rated on a 5-point scale from Never to Always (Never, Rarely, Sometimes, Most of the time, Always).
- Accuracy: How accurate participants perceived the AI tool's suggestions in supporting diagnosis, rated on a 5-point scale from Very inaccurate to Very accurate (Very inaccurate, Somewhat inaccurate, Neutral, Accurate, Very accurate).
- Efficiency: Whether the AI tool improved the diagnostic process, rated on a 5-point scale from Slowed significantly to Improved significantly (Slowed significantly, Slowed slightly, No impact, Improved slightly, Improved significantly).
- Trust: The degree to which participants trusted the AI tool's suggestions and guidance for clinical decision-making, rated on a 5-point scale from Never to Completely (Never, Rarely, Somewhat, Mostly, Completely).

Summary:

- Usability: Most participants found the tool easy or very easy to use (7/10).
- Relevance of Responses: Six out of ten reported the responses as mostly or always relevant.
- Accuracy: Seven out of ten rated the tool as accurate or very accurate.
- Efficiency: Eight out of ten felt the tool slightly or significantly improved their diagnostic process.
- Trust: Six out of ten reported mostly or complete trust in the AI suggestions.

https://doi.org/10.3390/electronics15030555

Table 14. Participant feedback on KG-enhanced LLM (Likert scale responses).

ID	Usability	Relevance of Responses	Accuracy	Efficiency	Trust
DR01	Easy	Most of the time	Neutral	Improved slightly	Mostly
DR02	Easy	Most of the time	Accurate	Improved significantly	Completely
DR03	Very easy	Always	Very accurate	Improved significantly	Completely
DR04	Neutral	Always	Very accurate	Slowed slightly	Mostly
DR05	Difficult	Rarely	Somewhat inaccurate	Slowed slightly	Somewhat
DR06	Very easy	Always	Very accurate	Improved slightly	Mostly
DR07	Neutral	Most of the time	Accurate	Improved slightly	Mostly
DR08	Neutral	Sometimes	Neutral	Improved slightly	Somewhat
DR09	Neutral	Sometimes	Neutral	Improved slightly	Somewhat
DR10	Difficult	Sometimes	Accurate	Improved slightly	Somewhat

5. Discussion

PHP is an endocrine disorder that is frequently misdiagnosed, particularly as epilepsy in some regions, because of overlapping neurological symptoms. Diagnostic complexity is compounded by the condition's rarity and the varied presentations of its subtypes.

5.1. Case Study 1 (Typical Presentation: PHP Type 1A)

In Case 1 (Table 7), which included clearly abnormal laboratory values, diagnoses in the non-AI round were broad and inconsistent. Suggestions included secondary hypoparathyroidism, Cushing's syndrome, adrenal insufficiency, and in some instances, no definitive diagnosis. Most participants ultimately opted to refer the case. Four participants initially suspected Cushing's syndrome due to truncal obesity associated with AHO; one also mentioned osteodystrophy. One participant referred the patient to a neurologist because of seizure-like features. Only one participant, a junior resident with recent endocrinology experience, correctly diagnosed pseudohypoparathyroidism, but did not specify a subtype.

With AI support, participants engaged more effectively with the case. Six reached a diagnosis of PHP without specifying the subtype, whereas one participant asked targeted questions and used the KG-enhanced LLM to identify the correct subtype. Three participants remained inconclusive despite AI assistance.

5.2. Case Study 2 (Atypical Presentation: Pseudo-PHP)

Without AI assistance, most participants struggled with Case 2 (Table 8). The first participant was unable to make a diagnosis, even with the KG-enhanced LLM, because the information provided was deemed unhelpful. Among the subsequent eight participants, several requested genetic testing but were unfamiliar with how to interpret the results. Four offered incorrect differentials, Down syndrome, DiGeorge syndrome, autism, or Cushing's syndrome, based on physical features and observed behavioral abnormalities. Most opted to refer to a specialist (neurologist or paediatrician). However, only the last participant, the junior resident, correctly referred to an endocrinologist.

With AI support, participants navigated the atypical presentation more effectively. Four participants correctly diagnosed pseudo-PHP, while two identified PHP without specifying a subtype. Four participants remained inconclusive or misdiagnosed the case. Among the nine out of ten participants who completed Case 1 before Case 2, those who identified PHP without specifying a subtype found Case 2 confusing because its presentation resembled Case 1 but with normal laboratory values. This prompted some participants to ask more targeted questions, which in some cases led to identifying the specific subtype, something they had not achieved in Case 1.

https://doi.org/10.3390/electronics15030555

5.3. Case Study 3 (Control: Severe Malaria)

In Case 3, most participants relied on their own clinical judgement. Without AI, only four out of ten participants diagnosed malaria, and just one correctly specified severe malaria. The majority suspected alternative diagnoses such as sepsis, pneumonia, or metabolic acidosis. Given the clinical findings, including a high white blood cell count and an abnormal anion gap, sepsis or metabolic acidosis were not unreasonable. Similarly, pneumonia was suspected because of respiratory distress.

Seven out of ten participants expressed high confidence in their clinical assessment, noting relief at handling a familiar condition and choosing not to use the KG-enhanced LLM. The three who engaged with the AI found it unhelpful because the clinical features of this case were not represented in the KG.

5.4. Participant Reflections and Feedback

Most participants noted that the interview felt more like an exam, which made it easier to forget routine questions and omit standard investigations, e.g. failing to request a malaria parasite test in Case 3, which is routine in the region). Participants also emphasised that, in clinical practice in this region, making a precise diagnosis is not always the immediate priority. Instead, the focus is often on managing presenting complaints and clinical abnormalities before referring the patient to a specialist. Most participants appreciated that the system provided information only when asked, rather than offering unsolicited suggestions; this gave users a sense of control and reduced the risk of information overload. However, some participants were concerned that the KG-enhanced LLM often provided too much information to be practical in a clinical setting, reflecting the context precision score (0.76) and its implications for cognitive load. "Responses need to be more specific," one participant emphasised. This design also placed an additional burden on clinicians, who had to know what to ask and how to ask it. Participants uncertain about next steps or terminology sometimes failed to uncover helpful leads, not because the AI lacked the answer, but because the prompt did not provide practical guidance.

Many participants emphasised the importance of transparency and trustworthiness in medical AI tools. Participants suggested validation mechanisms, such as tracking accuracy rates or implementing clinical trials, before full adoption. Others proposed domain-specific restrictions or safeguards, although some could not identify specific requirements, possibly due to unfamiliarity with AI regulations or limitations. Several participants expressed concern that clinicians might gradually trust AI tools more than their own diagnostic reasoning, potentially leading to a decline in critical thinking over time. This aligns with existing concerns in the literature about automation bias in medical decision-making.

Participant feedback indicated a generally positive reception toward AI-assisted tools, particularly as supportive instruments rather than primary diagnostic tools. Most participants would consider incorporating such a tool into their workflow, especially for rare or complex cases, but not in emergencies or routine scenarios where clinical judgment is more straightforward. Some participants feel the tool had greater value as an educational aid than as a primary diagnostic tool.

A relevant question that emerged, although not explicitly asked, seems crucial: Do clinicians fear misdiagnosing patients more because of their own judgment or because of over-reliance on an AI tool? This distinction could provide deeper insight into how responsibility, confidence, and trust interact in clinical decision-making. Future evaluations should incorporate such a question.

https://doi.org/10.3390/electronics15030555

5.5. Limitations

This study has several limitations, the most significant being the small number of participants and the limited evaluation dataset. Only ten clinical questions were used, and the study involved ten participants. These constraints limit generalisability, and the results should be interpreted as exploratory rather than conclusive. While the qualitative insights were rich and meaningful, they do not support statistically significant conclusions.

Several limitations relate to the KG-enhanced LLM framework and its supporting knowledge graph. The KG was manually curated from trusted sources such as textbooks and clinical guidelines. While reliable, this introduced biases affecting completeness and scope. Selection bias occurred because only well-documented information about PHP and epilepsy was included, whereas newer or less established findings were excluded, limiting representation of the full clinical picture. Expert bias also influenced content, reflecting the perspectives and priorities of its creator. For example, the KG may emphasise certain causes, such as low calcium in PHP-related seizures, while overlooking alternative explanations, including neurological conditions or atypical presentations. This narrowing of diagnostic paths could reduce the likelihood of surfacing rare but important differentials. The restricted scope of the KG further limited its utility: with only two diseases represented, it could not provide detailed differentials or address broader diagnostic queries, and symptoms were represented simply as present or absent, without considering severity, frequency, or triggers. This simplification could reinforce textbook-style reasoning and, in complex cases, bias participants toward familiar presentations.

Other limitations relate to the clinical evaluation itself. Participant experience levels were skewed: nine had 1–3 years of clinical experience, and one had 3–5 years. All were general practitioners except for one junior resident. While this offered consistency in perspective, it reduced diversity in clinical backgrounds and may have influenced how the KG-enhanced LLM was used and evaluated. The simulated diagnostic setting was also artificial; several participants noted that sessions felt more like exams than natural clinical interactions, which could have affected prompting style, communication confidence, and willingness to explore the system. Additionally, the order of case presentation was not fully balanced, potentially introducing fatigue, priming effects, or familiarity bias. Because participants completed the same cases in both the AI-assisted and non-AI phases, observed differences may reflect learning effects rather than the AI's impact. Longer-term impacts, such as whether repeated use would influence diagnostic confidence, accuracy, or cognitive bias, were not assessed due to time constraints and study design.

Usability was also a constraint. The system relied entirely on participants to frame questions, offering no guidance when queries were vague or unclear. Consequently, participants sometimes failed to obtain useful responses even when the relevant information was present in the KG.

Despite these limitations, the study provides valuable exploratory insights into how clinicians interact with KG-enhanced LLMs and highlights practical challenges and considerations for future clinical evaluations, including participant diversity, naturalistic settings, and system usability.

5.6. Future Improvements

Future development of the KG-enhanced LLM should begin with targeted technical improvements. Expanding the KG's size and scope would enable coverage of a broader range of diseases, rare presentations, and atypical symptom patterns, supporting cases characterised by high diagnostic uncertainty. However, such expansion must be accompanied by careful information prioritisation and relevance filtering to prevent excessive or poorly structured information from overwhelming clinicians. Metric inconsistencies observed in

https://doi.org/10.3390/electronics15030555

the current evaluation highlight the need for additional safeguards. Future work could incorporate abstention or fallback mechanisms to explicitly signal uncertainty when context is insufficient, alongside systematic failure-case analyses. Introducing a re-ranking stage to prioritise clinically relevant entities and relationships could further improve retrieval quality, interpretability, and overall clinical reliability.

An expanded KG would also enable larger, clinician-led trials to more rigorously examine diagnostic reasoning in atypical and rare cases under increased information volume. Rather than treating AI responses as standalone answers, future studies should explicitly examine how retrieved information supports different stages of clinical reasoning, such as hypothesis generation, confirmation, exclusion of alternatives, and confidence calibration. Capturing these interaction patterns would clarify when KG-enhanced LLMs provide meaningful support and when they are bypassed due to low perceived need for assistance.

Such trials would additionally allow for the assessment of whether KG-enhanced LLMs meaningfully reduce diagnostic effort or instead introduce additional cognitive load. Iterative refinement of system prompts and interaction design could then be used to optimise response length, tone, usability, and cognitive load reduction, supporting real-world clinical adoption. Future evaluations should also systematically assess interaction efficiency under realistic time and workload constraints, including prompting behaviour such as the number, length, and specificity of prompts, as well as the proportion of clinically useful information returned. Prompt-tuning strategies, context-aware interactions, and proactive detection of vague or incomplete inputs—with suggested clarifying follow-up questions—may further reduce interaction friction and minimise trial-and-error during clinical reasoning.

To reliably isolate AI-specific effects in future studies, a cross-over design with randomised case order would be required. Where cases are repeated, inclusion of a washout period would help minimise learning effects from prior exposure, ensuring that observed improvements can be more confidently attributed to AI assistance. Future evaluations should also expand the number and diversity of out-of-KG cases and explicitly require participant interaction with the system to reliably assess hallucination rates, abstention behaviour, and handling of unsupported or unfamiliar queries. In parallel, bias mitigation strategies, including adversarial datasets designed to expose overfitting or spurious correlations, will be essential to prevent misleading or overly narrow diagnostic suggestions. Evaluation frameworks should further account for cultural and clinical practice variations, as well as ethical considerations surrounding responsibility, trust, and accountability in clinical decision-making.

Long-term integration goals focus on embedding the KG-enhanced LLM in ways that align with clinicians' workflows and decision-making practices. Multi-centre validation across diverse clinical settings will be important to understand how clinicians adopt the system selectively, identify workflow-specific constraints, and evaluate usability in real-world contexts. Robust APIs compliant with interoperability standards such as HL7 FHIR would support smooth data exchange and context-aware decision support, reducing friction for clinicians. Piloting the system in targeted clinical settings will provide insight into clinician interaction patterns, including frequency of use, number and type of prompts issued per case, and how AI input is balanced with professional judgment. Collecting continuous feedback from clinicians will guide iterative refinements, ensuring that the system supports efficient decision-making, maintains trust, and integrates safely into routine practice without adding undue cognitive burden.

https://doi.org/10.3390/electronics15030555

6. Conclusions

This study demonstrates that a KG-enhanced LLM can effectively support clinicians in complex or rare cases, particularly those with atypical presentations, while offering limited benefit in routine or familiar scenarios. Rather than replacing clinical judgment, the system functioned as an assistive tool, supporting reasoning, providing second-opinion insights, and acting as an educational aid. Clinicians were more likely to engage with AI support when diagnostic confidence was low, especially in rare endocrine cases such as PHP. Notably, AI responses that explicitly acknowledged uncertainty increased clinician trust, suggesting that transparency and humility are important design features for medical AI.

The findings also show that the system's usefulness depends heavily on clinician interaction, requiring users to recognise uncertainty and articulate effective queries. Improving the model's ability to detect ambiguity and proactively guide users through prompt suggestions may enhance its clinical value. Participants consistently preferred to rely on their own judgment in familiar cases, indicating that KG-enhanced LLMs are most beneficial in situations characterised by diagnostic uncertainty rather than routine decision-making.

Safeguards remain essential to prevent overreliance, preserve clinical reasoning, and reduce the risk of misdiagnosis, reinforcing the need for AI to remain an assistive, not authoritative, component of clinical workflows. Strengthening the underlying KG and validating performance across larger and more diverse datasets will be critical for ensuring reliability. Transparent feedback and performance indicators may further support trust and responsible adoption.

With careful design and validation, KG-enhanced LLMs can serve as effective collaborative tools in clinical decision-making, enhancing diagnostic confidence while keeping final responsibility with clinicians.

Author Contributions: Conceptualization, F.S. and J.W.; methodology, F.S.; software, F.S.; validation, F.S.; formal analysis, F.S.; investigation, F.S.; data curation, F.S.; writing—original draft preparation, F.S.; writing—review and editing, F.S. and J.W.; visualization, F.S.; supervision, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was approved by the Ethical Committee of the University of West London.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The original contributions presented in this study are included in the article. The backend, frontend, and knowledge graph dump are publicly available on GitHub at https://github.com/fateeS88/MKG-Pseudohypoparathyroidism-.git (accessed on 20 December 2025). Further enquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AHO	Albright Hereditary Osteodystrophy
CDSS	Clinical Decision Support Systems
EHRs	Electronic Health Records
EMRs	Electronic Medical Records
GDPR	General Data Protection & Regulation
GOPD	General Outpatient Departments
GP	General Practitioner
KG	Knowledge Graph

LLM	Large Language Model
MKG	Medical Knowledge Graphs
MMR	Maximal Marginal Ranking
NICE	National Institute for Health and Care Excellence
PHP	Pseudohypoparathyroidism
PLM	Pre-trained Language Model
PTH	Parathyroid Hormone
Q&A	Question-Answering
RAG	Retrieval-Augmented Generation
RAGAS	Retrieval-Augmented Generation Assessment System
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RR	Re-Ranking
SD	Standard Deviation
SFT	Supervised Fine-Tuning
SQuAD	Stanford Question Answering Dataset
UHC	Universal Health Coverage
UMLS	Unified Medical Language System
USMLE	United States Medical Licensing Examination

Appendix A. Prompt Template

The following prompt template was used to generate responses from the KG-enhanced LLM. Retrieved knowledge graph context was injected verbatim, and the model was instructed to ground its response strictly in the provided information.

Knowledge Graph Context:

{Retrieved triples formatted as subject–relation–object statements}

User Question:

{User-provided clinical query}

Instruction:

Please answer using only the information from the knowledge graph context above. If the information required to answer the question is not available, respond with *"I couldn't find this information in my knowledge base."*

References

1. Medford-Davis, L.; Malani, R. The Physician Shortage Isn't Going Anywhere. 2024. Available online: https://www.mckinsey.com/industries/healthcare/our-insights/the-physician-shortage-isnt-going-anywhere (accessed on 17 December 2025).
2. Sutton, R.; Pincock, D.; Baumgart, D.; Sadowski, D.; Fedorak, R.; Kroeker, K. An overview of clinical decision support systems: Benefits, risks, and strategies for success. *npj Digit. Med.* **2020**, *3*, 17. [CrossRef]
3. Nazi, Z.A.; Peng, W. Large language models in healthcare and medical domain: A review. *Informatics* **2024**, *11*, 57. [CrossRef]
4. Zou, S.; He, J. Large Language Models in Healthcare: A Review. In Proceedings of the 2023 7th International Symposium on Computer Science and Intelligent Control (ISCSIC), Nanjing, China, 27–29 October 2023. [CrossRef]
5. Jovanović, M.; Campbell, M. Connecting AI: Merging Large Language Models and Knowledge Graph. *Computer* **2023**, *56*, 103–108. [CrossRef]
6. Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3580–3599. [CrossRef]
7. Kim, E.; Shrestha, M.; Foty, R.; DeLay, T.; Seyfert-Margolis, V. Structured Extraction of Real World Medical Knowledge using LLMs for Summarization and Search. *arXiv* **2024**, arXiv:2412.15256. [CrossRef]
8. Abu-Salih, B. Domain-specific knowledge graphs: A survey. *J. Netw. Comput. Appl.* **2021**, *185*, 103076. [CrossRef]
9. Cai, L.; Yu, C.; Kang, Y.; Fu, Y.; Zhang, H.; Zhao, Y. Practices, opportunities and challenges in the fusion of knowledge graphs and large language models. *Front. Comput. Sci.* **2025**, *7*, 1590632. [CrossRef]

10. Jia, M.; Duan, J.; Song, Y.; Wang, J. medIKAL: Integrating Knowledge Graphs as Assistants of LLMs for Enhanced Clinical Diagnosis on EMRs. In Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, United Arab Emirates, 19–24 January 2025; Association for Computational Linguistics: Stroudsburg, PA, USA , 2025; pp. 9278–9298. Available online: https://aclanthology.org/2025.coling-main.624/ (accessed on 28 November 2025).
11. Park, C.; Lee, H.; Lee, S.; Jeong, O. Synergistic Joint Model of Knowledge Graph and LLM for Enhancing XAI-Based Clinical Decision Support Systems. *Mathematics* **2025**, *13*, 949. [CrossRef]
12. Newton, N.; Bamgboje-Ayodele, A.; Forsyth, R.; Tariq, A.; Baysari, M.T. How Are Clinicians' Acceptance and Use of Clinical Decision Support Systems Evaluated Over Time? A Systematic Review. In *MEDINFO 2023—The Future Is Accessible*; Studies in Health Technology and Informatics; IOS Press: Amsterdam, The Netherlands, 2024. [CrossRef]
13. Westerbeek, L.; Ploegmakers, K.J.; de Bruijn, G.-J.; Linn, A.J.; van Weert, J.C.M.; Daams, J.G.; van der Velde, N.; van Weert, H.C.; Abu-Hanna, A.; Medlock, S. Barriers and facilitators influencing medication-related CDSS acceptance according to clinicians: A systematic review. *Int. J. Med. Inform.* **2021**, *152*, 104506. [CrossRef]
14. Xie, Q.; Schenck, E.J.; Yang, H.S.; Chen, Y.; Peng, Y.; Wang, F. Faithful AI in Medicine: A Systematic Review with Large Language Models and Beyond . *medRxiv* **2023**. [CrossRef]
15. Yang, R.; Liu, H.; Marrese-Taylor, E.; Zeng, Q.; Ke, Y.; Li, W.; Cheng, L.; Chen, Q.; Caverlee, J.; Matsuo, Y.; et al. KG-Rank: Enhancing Large Language Models for Medical QA with Knowledge Graphs and Ranking Techniques. In Proceedings of the 23rd Workshop on Biomedical Natural Language Processing, Bangkok, Thailand, 16 August 2024; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024. [CrossRef]
16. Yang, R.; Tan, T.R.; Lu, W.; Thirunavukarasu, A.J.; Shu, D.; Liu, N. Large Language Models in Health Care: Development, Applications, and Challenges. *Health Care Sci.* **2023**, *2*, 255–263. [CrossRef]
17. Balogh, E.P.; Miller, B.T.; Ball, J.R. Overview of Diagnostic Error in Health Care. In *Improving Diagnosis in Health Care*; National Academies Press: Washington, DC, USA, 2019. Available online: https://www.ncbi.nlm.nih.gov/books/NBK338594/ (accessed on 17 December 2025).
18. Sadat, M.; Zhou, Z.; Lange, L.; Araki, J.; Gundroo, A.; Wang, B.; Menon, R.; Parvez, M.R.; Feng, Z. DELUCIONQA: Detecting Hallucinations in Domain-specific Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 822–835. Available online: https://aclanthology.org/anthology-files/anthology-files/pdf/findings/2023.findings-emnlp.59.pdf (accessed on 19 September 2025).
19. Pan, S.; Zheng, Y.; Liu, Y. Integrating Graphs With Large Language Models: Methods and Prospects. *IEEE Intell. Syst.* **2024**, *39*, 64–68. [CrossRef]
20. Ibrahim, N.; Aboulela, S.; Ibrahim, A.; Kashef, R. A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): Models, evaluation metrics, benchmarks, and challenges. *Discover AI* **2024**, *4*, 76. [CrossRef]
21. Yadav, D.; Para, H.; Selvakumar, P. Unleashing the Power of Large Language Model, Textual Embeddings, and Knowledge Graphs for Advanced Information Retrieval. In Proceedings of the 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 16–17 November 2023. [CrossRef]
22. Zhang, W.; Tian, Y.; Meng, X.; Wang, M.; Du, J. Knowledge Graph-Infused Fine-Tuning for Structured Reasoning in Large Language Models. *arXiv* **2025**, arXiv:2508.14427. [CrossRef]
23. Gao, Y.; Li, R.; Croxford, E.; Caskey, J.; Patterson, B.W.; Churpek, M.; Miller, T.; Dligach, D.; Afshar, M. Leveraging Medical Knowledge Graphs Into Large Language Models for Diagnosis Prediction: Design and Application Study. *JMIR AI* **2025**, *4*, e58670. [CrossRef] [PubMed]
24. Croxford, E.; Gao, Y.; Patterson, B.; To, D.; Tesch, S.; Dligach, D.; Mayampurath, A.; Churpek, M.M.; Afshar, M. Development of a human evaluation framework and correlation with automated metrics for natural language generation of medical diagnoses (Preprint). *medRxiv* **2024**. [CrossRef]
25. Shamszare, H.; Choudhury, A. Clinicians' Perceptions of Artificial Intelligence: Focus on Workload, Risk, Trust, Clinical Decision Making, and Clinical Integration. *Healthcare* **2023**, *11*, 2308. [CrossRef]
26. Bastepe, M.; Gensure, R.C. Hypoparathyroidism and Pseudohypoparathyroidism. [Updated 2024 May 8]. In *Endotext [Internet]*; Feingold, K.R., Adler, R.A., Ahmed, S.F., Anawalt, B., Blackman, M.R., Chrousos, G., Corpas, E., de Herder, W.W., Dhatariya, K., Dungan, K., et al., Eds.; MDText.com, Inc.: South Dartmouth, MA, USA 2000. Available online: https://www.ncbi.nlm.nih.gov/sites/books/NBK279165/ (accessed on 17 December 2025).
27. Najim, M.S.; Mohammed, A.; Awad, M.; Ahmed, O. Pseudohypoparathyroidism presenting with seizures: A case report and literature review. *Intractable Rare Dis. Res.* **2020**, *9*, 166–170. [CrossRef]
28. Rodriguez, J.A.; Roa, A.A.; Leonso-Bravo, A.-A.; Khatiwada, P.; Eckardt, P.; Lemos-Ramirez, J. A Case of Plasmodium falciparum Malaria Treated with Artesunate in a 55-Year-Old Woman on Return to Florida from a Visit to Ghana. *Am. J. Case Rep.* **2020**, *21*, e926097. [CrossRef] [PubMed]
29. Dai, L.-Z.; Lin, C.; Lei, R.; Zhang, Y.; Ma, H. A Case of Pseudohypoparathyroidism Misdiagnosed as Idiopathic Epilepsy for 5 Years: Clinical Analysis and Follow-up Outcomes. *J. Int. Med. Res.* **2023**, *51*. [CrossRef]

30. OpenAI. OpenAI Platform. 2025. Available online: https://platform.openai.com/docs/models/compare (accessed on 17 December 2025).
31. Dhanakotti, K. RAGAS for RAG in LLMs: A Comprehensive Guide to Evaluation Metrics. Medium. 2024. Available online: https://dkaarthick.medium.com/ragas-for-rag-in-llms-a-comprehensive-guide-to-evaluation-metrics-3aca142d6e38 (accessed on 17 December 2025).
32. Ghaffar, F.; Furtado, N.M.; Ali, I.; Burns, C. Diagnostic Decision-Making Variability Between Novice and Expert Optometrists for Glaucoma: Comparative Analysis to Inform AI System Design. *JMIR Med. Inform.* **2025**, *13*, e63109. [CrossRef] [PubMed]
33. Intersoft Consulting. General Data Protection Regulation (GDPR). 2018. Available online: https://gdpr-info.eu/issues/consent/ (accessed on 17 December 2025).
34. NICE. 1 Recommendations. In *Artificial Intelligence (AI)-Derived Software to Help Clinical Decision Making in Stroke*; NICE: London, UK, 2024. Available online: https://www.nice.org.uk/guidance/dg57/chapter/1-Recommendations (accessed on 17 December 2025).
35. NICE. 3 Committee Discussion. In *Artificial Intelligence (AI)-Derived Software to Help Clinical Decision Making in Stroke*; NICE: London, UK, 2024. Available online: https://www.nice.org.uk/guidance/dg57/chapter/3-Committee-discussion (accessed on 17 December 2025).
36. EY Global. G7 AI Principles and Code of Conduct. 2023. Available online: https://www.ey.com/en_gl/insights/ai/g7-ai-principles-and-code-of-conduct (accessed on 17 December 2025).
37. Neha, F.; Bhati, D.; Shukla, D.K. Retrieval-Augmented Generation (RAG) in Healthcare: A Comprehensive Review. *AI* **2025**, *6*, 226. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Implementing a Knowledge Management System with GraphRAG: A Physical Internet Example

Hisatoshi Naganawa [1,*], Enna Hirata [2] and Akira Yamada [3]

[1] International Maritime Research Center, Kobe University, Kobe 658-0022, Japan
[2] Graduate School of Maritime Sciences, Kobe University, Kobe 658-0022, Japan; enna.hirata@platinum.kobe-u.ac.jp
[3] Graduate School of Information Science, University of Hyogo, Kobe 650-0047, Japan; akirayamada@acm.org
* Correspondence: 2117053z@gsuite.kobe-u.ac.jp

Abstract: The rapid expansion and interdisciplinary nature of Physical Internet (PI) research have resulted in fragmented knowledge, limiting the ability of stakeholders to identify emerging trends, actionable insights and genuine research gaps. This study introduces a novel knowledge management approach that uses Graph Retrieval-Augmented Generation (GraphRAG) to systematically organize and integrate PI-related literature. A comprehensive knowledge graph was constructed by extracting and semantically modeling entities and relationships from 2835 academic papers, conference proceedings and international roadmaps. The developed system incorporates fuzzy semantic search and multiple retrieval strategies, including local, global and hybrid approaches, enabling nuanced, context-aware access to information. Stakeholder-specific prompts, tailored to the needs of industry, government and academia, demonstrate how GraphRAG can support the discovery of business model innovations, policy design and underexplored research areas. A comparative evaluation using cosine similarity and BERTScore confirms that graph-based strategies outperform standard LLM retrieval in providing relevant and comprehensive answers while also revealing connections that would be missed in manual reviews. The results demonstrate that the proposed GraphRAG model is a scalable and extensible framework for addressing knowledge gaps and promoting collaboration in PI research synthesis for sustainable logistics. The model also shows promise for application in other complex domains.

Keywords: GraphRAG; knowledge management; physical internet; LLM

1. Introduction

The logistics industry is facing increasing challenges, particularly in addressing its significant contribution to global greenhouse gas (GHG) emissions. These emissions are often exacerbated by operational inefficiencies such as suboptimal routing, inefficient warehouse layouts, and poor inventory management [1]. In response to these challenges, the concept of the PI has emerged as a transformative framework. Introduced by Montreuil et al. [2], PI reimagines global logistics networks by applying design principles from the digital internet to the transportation and distribution of physical goods. By leveraging information and communication technologies (ICT) and standardized logistics infrastructures, PI aims to enhance efficiency, flexibility, and sustainability in freight transportation systems. Over the past decade, international conferences like International Physical Internet Conference (IPIC) and government-led initiatives in regions such as Europe and Japan have accelerated discussions and roadmap developments aimed at realizing the PI vision.

Despite its potential, advancing research in the PI domain requires systematic methods for organizing and integrating diverse knowledge sources. The growing body of literature on PI—spanning academic research, government reports, and industry white papers—presents both opportunities and challenges for researchers seeking to identify trends, gaps, and actionable insights. Addressing this complexity calls for innovative tools that can manage and synthesize large volumes of information effectively. However, manual review articles on PI have reviewed less than 700 articles, e.g., 663 articles in [3], yet there are more than 2000 scientific papers already published.

Despite there being several systematic reviews and bibliometric studies, the synthesis of current knowledge in the PI field is severely hindered by fragmentation, methodological heterogeneity and the rate at which new research is produced. As noted by Tallaki et al., previous manual reviews only analyzed a subset of available publications (less than 700), leaving significant gaps due to resource constraints and limited semantic scope [4]. To address this challenge, we present the Graph Retrieval-Augmented Generation (GraphRAG) method. By building a comprehensive, semantically rich knowledge graph, GraphRAG enables information to be integrated and retrieved from thousands of diverse sources. This scalable, flexible framework for knowledge organization fills a critical gap left by traditional review methods.

GraphRAG is an emerging Artificial intelligence (AI)-driven approach that combines natural language understanding with graph-based relational structures to enhance information retrieval and generation. Unlike traditional methods, GraphRAG excels at uncovering relationships among data points, retrieving highly relevant information, and generating contextually rich outputs. These capabilities make it particularly well-suited for addressing complex problems with interconnected data structures, such as those found in PI research.

This study explores the application of GraphRAG as a novel method for advancing knowledge management in the PI domain. Specifically, it constructs a knowledge graph using GPT-4o mini and Neo4j to systematically analyze PI-related literature. By integrating fragmented research findings, visualizing emerging trends, and identifying knowledge gaps, this approach aims to deepen understanding and foster innovation in sustainable logistics systems.

This study makes three key contributions to the literature on knowledge management and sustainable logistics innovation. First, it represents the first known application of GraphRAG to construct a comprehensive knowledge graph for the PI domain. This graph integrates 2835 documents including academic articles, IPIC proceedings, and strategic roadmaps from the EU and Japan, enabling a structured and scalable analysis of emerging trends, research gaps, and innovation opportunities.

Second, it introduces a fuzzy matching mechanism to improve semantic search within the knowledge graph. By applying cosine similarity with a threshold-based retrieval method, the system improves performance in handling ambiguous queries, such as "sustainable logistics policies", achieving a reported precision of 87%.

Third, the study designs and evaluates stakeholder-specific prompts tailored to the needs of industry practitioners, academic researchers, and policymakers. These prompts are assessed across four retrieval strategies, namely GPT-only, local, global, and hybrid, demonstrating the model's ability to deliver actionable insights grounded in both node-level detail and community-level context.

The remainder of this paper is structured as follows: Section 2 reviews relevant literature on PI and generative AI technologies like GraphRAG. Section 3 describes the data sources, and Section 4 presents the methodology used to construct the knowledge graph. Section 5 presents results from various search strategies and evaluates their performance. Finally, Section 6 concludes with key findings and identifies future research directions.

2. Literature Review

2.1. Physical Internet

The concept of PI was first introduced by Montreuil et al. as a transformative framework for sustainable logistics [2]. By applying the design principles of the digital Internet, PI aims to create an interoperable, collaborative and efficient freight transport system. The core idea is to route modular containers through transit hubs, called PI hubs, to maximize consolidation opportunities and minimize inefficiencies in transportation networks [4]. Over the past decade, PI has received considerable attention from researchers and practitioners alike for its potential to address critical challenges such as greenhouse gas emissions and resource inefficiencies in global logistics systems [5].

The growing body of literature on PI reflects its increasing importance. The growing academic interest in the PI is reflected in the increasing volume of peer-reviewed publications and systematic literature reviews over the past decade, as documented by recent studies [6–8]. Foundational works by Ballot et al., Aron and Sgarbossa, and Wu et al. have systematically reviewed PI research, summarized its progress, and identified future research directions.

Existing review papers on PI have only surveyed a limited number of studies. For example, Tallaki et al. surveyed 663 publications using bibliometric methods [4]. Nguyen et al. employed a bibliometric knowledge mapping approach to assess research landscape in PI, digital twin and supply chain management areas [9]. In this study, 467 papers were reviewed. These examples highlight the need for a more comprehensive review to a larger or complete set of relevant literature to achieve a thorough and holistic synthesis of the current state of the art.

Despite these efforts, the rapid expansion of PI-related research has created a fragmented knowledge base. This fragmentation stems not merely from an information overload due to the sheer increase in the number of publications but from the interplay of multiple factors. First, methodological inconsistencies are a key factor. Existing review studies exhibit significant differences in the literature databases they utilize, their analytical methodologies (e.g., systematic reviews, bibliometric analysis, content analysis), and the metrics they employ. Consequently, the comparability of research findings is low, and the insights gained tend to be partial and limited. Second, the dispersion and diversification of research subjects also contribute to fragmentation. PI research has developed while intersecting with diverse fields such as logistics system design, information and communication technology, sustainability, policy and regulation, and business models. Consequently, individual researchers focus on different themes and analytical units, hindering sufficient integration of cross-disciplinary knowledge. Third, insufficient theoretical integration is a significant factor. Much existing PI research relies on individual case studies or technical approaches, with insufficient systematization of theoretical models or conceptual frameworks as an academic foundation. This makes it difficult to understand research findings in relation to each other, leading to a tendency for insights to remain scattered.

Considering these challenges, this study aims to reconstruct the overall landscape of PI research from a more comprehensive and integrated perspective, clarifying the structure and developmental trends of the research field.

2.2. Generative AI

AI has attracted significant attention in a variety of fields and industries. Beginning with Alan Turing's foundational ideas in the 1950s, AI has undergone multiple phases of development, including periods of stagnation famously referred to as AI winters [10]. Generative AI technologies have revolutionized information processing across various domains. Since the release of OpenAI's ChatGPT-3.5 in 2022, large language models (LLMs)

have exhibited exceptional capabilities to understand context and generate text that closely resembles human language [11]. However, LLMs face limitations such as knowledge rigidity, reliance on static training data, and susceptibility to generating inaccurate or misleading outputs when confronted with unfamiliarity [12–14]. These challenges highlight the need for hybrid approaches that combine LLMs with external databases to enhance reliability and contextual accuracy.

2.3. GraphRAG

GraphRAG is an emerging technology that addresses the limitations of traditional LLMs by integrating natural language understanding with graph-based relational structures (see Figure 1). Introduced by Microsoft researchers in early 2024, GraphRAG improves information retrieval by linking LLMs to external knowledge graphs, enabling more accurate and contextually relevant responses [15,16]. Unlike similar technologies such as HybridRAG or LightRAG, GraphRAG excels at capturing indirect relationships between data points while maintaining high retrieval accuracy.

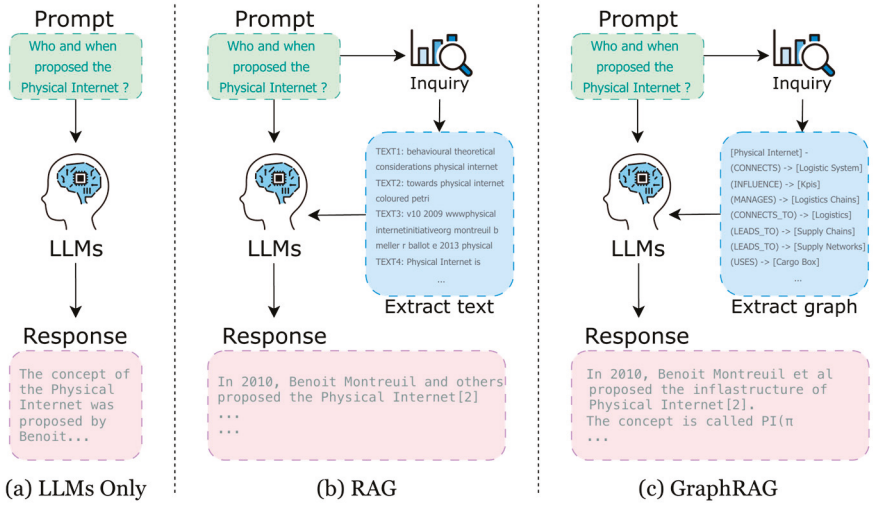


Figure 1. Different methods using LLM.

This diagram outlines an example where each plot related to PI is derived from the paper authored by Montreuil et al. [2].

Recent studies have begun to explore GraphRAG's applications in logistics and supply chain management. For example, Li et al. demonstrated its utility in transforming unstructured supplier data into knowledge graphs to improve supplier discovery processes [17]. Similarly, Ojima et al. proposed a GraphRAG-based knowledge management system for automotive failure analysis [18]. In the context of PI, Naganawa and Hirata presented a case study showing how GraphRAG can enhance sustainability efforts by enhancing policy generation [19].

Given its ability to integrate diverse data sources and uncover complex relationships, GraphRAG is a promising tool to address the fragmented knowledge landscape in PI research. This study builds on these advances by using GraphRAG to construct a comprehensive knowledge graph for PI-related literature, enabling systematic analysis of trends, gaps, and opportunities.

Technologies similar to GraphRAG are HybridRAG and LightRAG. We summarize the differences in these technologies in Table 1.

Overall, hybridRAG combines vector search and keyword search, enabling it to obtain data with strong superficial relevance, but it is difficult to capture indirect relationships. In addition, LightRAG is basically a vector search based on simple similarity and does

not consider relationships, so although it is fast in generation, there is a high possibility that fragmentary answers will be mixed in the generated results. Therefore, we considered GraphRAG to be the best choice for this study.

Table 1. Comparison of GraphRAG, HybridRAG, and LightRAG.

Feature	GraphRAG	HybridRAG	LightRAG
Retrieval Method	Graph-based plus semantic search	Combination of vector search and keyword search	Pure vector-based retrieval
Structure Awareness	Captures direct and indirect relationships via graph topology	Limited. Sometimes captures superficial relevance	Low. Relationships are not structurally modeled
Response Quality	High contextual depth with relational grounding	Moderate. May miss deeper or indirect context	Fast but may include fragmented or shallow responses
Scalability	Moderate. Depends on graph complexity	High. Simple integration of vector and keyword methods	Very high. Minimal processing overhead
Semantic Matching	Strong by fuzzy matching, cosine similarity	Moderate	High when using dense embeddings
Best Suited For	Integrated knowledge management, tailored stakeholder insights	General-purpose retrieval tasks	Quick. High-speed applications with simple queries

The fragmentation of PI literature is primarily due to the rapid growth in publications, interdisciplinary cross-linkages (e.g., between logistics, ICT and policy) and the lack of a standardized taxonomy or ontology. As topics, terminology and research agendas diverge, siloed knowledge clusters emerge, impeding cumulative learning. Our graph-based approach overcomes this issue by mapping explicit and latent relationships across domains, which traditional manual reviews are unable to achieve on a large scale.

3. Data

To construct a Knowledge Graph (KG) specifically tailored to the PI, this study aggregated data from four key sources (Table 2).

Table 2. List of Contents.

Category	Year	Number of Documents	Collection Method
Research Paper	2006–2024	2537	Semantic scholar API
Proceedings of IPIC	2016–2024	296	Download from conference repository
PI Roadmap (ALICE)	2020	1	Download from ALICE website
PI Roadmap (Japan)	2022	1	Download from METI website
Total		2835	

The first source consists of scientific literature. Using the keyword "Physical Internet", a total of 2537 relevant papers published between 2006 and 2024 were collected. We used the Semantic Scholar API to collect the publication year, author information, abstract, and full text. We chose 2006 as the starting point because it was the year the term "Physical Internet" was first used in the literature [20]. A single keyword, "Physical Internet," was adopted for keyword searches. This ensures the collection of all relevant literature and guarantees high comprehensiveness. Using multiple keywords simultaneously risks bias toward specific perspectives and could hinder comprehensive literature collection, making this the optimal choice for the objectives of this study.

The second source is conference proceedings of IPIC, papers published in IPIC proceedings from its inception in 2016 to the most recent year 2024, were collected. These proceedings offer valuable discussions on emerging trends, practical implementations, and collaborative efforts in PI research.

The third source is the European roadmap. The PI roadmap developed by the Alliance for Logistics Innovation through Collaboration in Europe (ALICE) was incorporated [21]. This roadmap outlines strategic initiatives and milestones aimed at advancing PI adoption in EU.

The fourth source is the national roadmap. The PI roadmap prepared by Japan's Ministry of Economy, Trade and Industry (METI) was also included [22]. This document provides a localized perspective on PI implementation strategies and priorities.

For data collection, the language was restricted to English. Exclusion criteria applied included removing duplicate documents with identical DOIs, excluding documents where the full text could not be obtained, and further excluding documents with low relevance to the keywords. These processes ensure that the collected documents and materials are highly reproducible and comprehensively cover information related to the PI domain.

4. Methodology

4.1. Preprocessing

The first step in constructing the KG for PI was to preprocess the raw data collected from four sources: scientific papers, IPIC proceedings, and existing roadmaps from EU and Japan were aggregated into a unified dataset. To ensure data quality, noise such as stop words, URLs, punctuation, and other irrelevant characters symbols removed using the Natural Language Toolkit (NLTK) and programing language Python 3.13.1. The cleaned and standardized corpus served as the input for subsequent graph construction. The overall preprocessing workflow is illustrated in Figure 2.

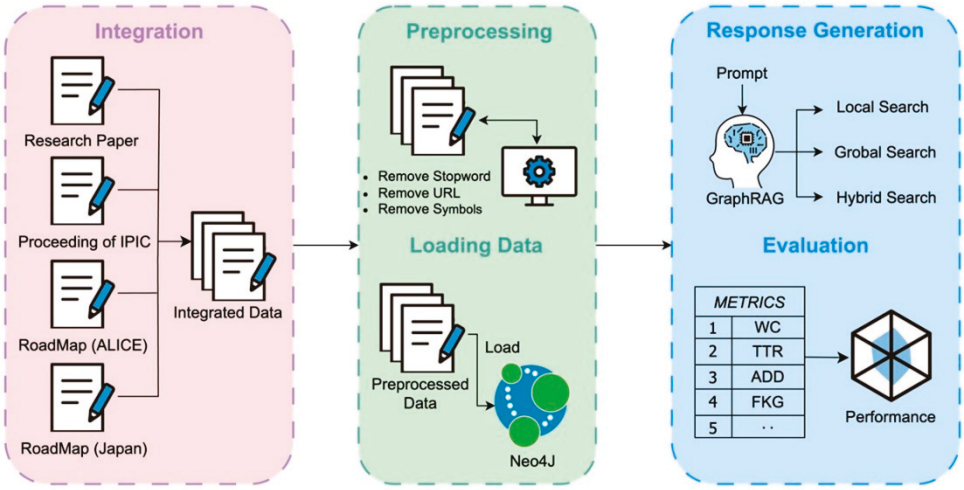


Figure 2. Data Processing Pipeline.

4.2. Knowledge Graph Construction

After preprocessing, the cleaned data was hosted in a Neo4j AuraDB graph database [23]. Using GPT-4o mini, entity and relationship extraction was conducted to form nodes and edges, which together created a structured knowledge graph representing the PI domain. Nodes typically represent key entities (e.g., "modular container," "sustainability policy"), while edges capture semantic or functional relationships between them. This structure enables flexible exploration of complex interconnections across heterogeneous information sources.

4.2.1. Entity and Relationship Extraction Process

The entity and relationship extraction process begins with document chunking. This involves dividing large documents into smaller, semantically coherent units, ensuring that

processing with language models remains manageable and contextually accurate. Common chunking techniques include token- or sentence-based splitting, with parameters configured to preserve topical integrity. Each chunk is assigned metadata such as the name of the parent document, its position and its length. These chunks are then represented as nodes within the knowledge graph and are linked to their parent documents to enable traceability.

Next comes entity and relationship extraction. Each chunk is passed to a large language model (LLM), such as GPT-4o mini, which is prompted to recognize entities (unique, meaningful concepts such as 'modular container' or 'sustainability policy') and relationships (semantic or functional connections such as 'enables', 'regulated by', or 'part of'). The LLM then outputs structured metadata for each entity, including its title, type, and description. It also outputs metadata for each relationship, including its source and target entities, the relationship type, and a description. Preprocessing is performed using NLP library NLTK for initial cleaning and noun phrase identification. The final definitions and descriptions are always reviewed or summarized using the LLM. The extraction prompt requests a list of all entities, along with their names, types, and descriptions, and the explicit relationships among them. This returns a structured JSON containing these details.

Following extraction, entity disambiguation and merging are performed to avoid redundant nodes in the final graph. Entities with similar names and types across multiple chunks are compared and merged where equivalent, and their descriptions are combined to ensure completeness.

During the node and edge definition stage, entities are represented as nodes characterized by their type and description. Chunks are also represented as nodes in order to preserve the structure and context of the document. Edges capture direct relationships between entities ('enables' or 'is-regulated-by', for example) and structural links such as 'PART_OF' or 'NEXT_CHUNK', which connect chunks to their parent documents or sequence them according to document order. Nodes and edges carry properties such as source chunk, document position, entity type or semantic summaries.

4.2.2. Semantic Modeling and Intermediate Steps

During semantic modeling and intermediate steps, nodes and edges are semantically annotated with domain-specific properties and aligned with a chosen ontology or domain schema to improve retrieval and reasoning. Domain-specific terminology and ontologies can guide more precise extraction by customizing LLM prompts to match the field's language. Additionally, each chunk, entity and relationship description is embedded using a sentence embedding model Paraphrase-MiniLM-L6-v2 [24]. This allows for semantic similarity searches and relevance ranking during retrieval. These embeddings are stored as properties on the nodes for querying in the future. The Paraphrase-MiniLM-L6-v2 model was selected for its high semantic textual similarity and computational efficiency when processing large document corpora. It was tested against other candidate embeddings.

Finally, all nodes and relationships, along with their semantic and structural properties, are ingested into Neo4j. The graph schema usually contains nodes labeled by entity types and relationships labeled by their semantic relationship types. When a query is submitted, GPT-4o mini is used again to translate the natural language question into a graph traversal query in Cypher (query language used in Neo4j) and to interpret and aggregate the relevant nodes and edges returned from the graph. This enables the generation of a natural language response or further reasoning based on the retrieved subgraphs. In the intermediate steps of this querying process, we apply fuzzy matching (see more details in Section 4.5) of query terms to entity nodes using embedding similarity. This process has adjustable thresholds to calibrate retrieval sensitivity.

4.3. Search Strategies

Four retrieval strategies were implemented and compared to evaluate their performance in generating insights. The four strategies are, I: GPT-only, II: Local Search, III: Global Search, and IV: Hybrid Search. The details are explained in Table 3. Hybrid Search is a strategy that combines "details" (Local) and "context" (Global) to generate more accurate and comprehensive knowledge.

Table 3. Outlines of searching methods applied in the model.

Method	Description
GPT-only	Normal LLM response generation (e.g., ChatGPT). GPT relies solely on the model's prior knowledge, without database augmentation.
Local Search	Search for nodes and edges in the graph that are semantically close to the query and retrieve directly related nodes and edges from the graph database. Local Search returns nodes and connected subgraphs that are directly relevant to the exact query.
Global Search	Identify communities (clusters) in the graph and obtain a summary of the entire community to which the query relates. Global Search retrieves broader, community-level graph segments to provide contextual breadth.
Hybrid Search	Integrates Local Search and Global Search to generate comprehensive responses with specific examples and background information. Hybrid Search combines node-level detail and global context, using reranking to achieve holistic relevance.

These strategies were designed to assess GraphRAG's ability to retrieve relevant information and generate accurate responses based on different levels of contextual depth.

4.4. Prompt Design

To preliminarily evaluate GraphRAG's usefulness under different user needs, we designed stakeholder-specific prompts for three groups: industry practitioners, policymakers, and academic researchers (Table 4). From an industry perspective, the focus was on identifying new business models enabled by PI. From a policymaker perspective, the goal was to explore policy approaches to promote global PI adoption. From an academic perspective, the aim was to discover unexplored areas within PI research.

Table 4. Overview of prompts.

No.	Perspective	Prompt
1	Industry practitioner	What new business models and innovations can be envisioned that leverage the Physical Internet?
2	Policymaker	What policy approaches should be taken to make the Physical Internet a reality in Global?
3	Academic researcher	What areas of research are missing that focus on the Physical Internet?

These prompts were executed using all four search strategies. Comparing the results provided initial insights into which approach could yield more comprehensive and actionable findings.

4.5. Parameter Tuning

We employed several parameters to optimize GraphRAG's performance. First, the temperature of the GPT-4o mini was set to 0 to ensure deterministic and consistent results. Second, the number of nodes referenced during response generation was capped at 50 to balance response speed and comprehensiveness. Third, we adopted fuzzy matching based on cosine similarity for the search algorithm, setting the threshold to 0.5. This threshold

is an empirical value based on preliminary experiments, determined by balancing the comprehensiveness and accuracy of search results. Cosine similarity measures the angle between the query's embedding vector and the node's embedding vector, enabling the capture of semantic similarity rather than exact string matches. For example, for the query "sustainable logistics policies," fuzzy matching also retrieves nodes related to "green logistics" and "carbon-neutral freight," enabling comprehensive coverage of related themes.

Equation (1) defines the relationship between temperature, softmax scaling, and vocabulary distribution in the language model.

$$p_i = \frac{exp^{y_{i/T}}}{\sum_j exp^{y_{j/T}}} \tag{1}$$

where P_i is a softmax function representing the probability of the ith class. Y_i is the logit for the ith token. T $(T > 0)$ is the LLM temperature parameter and n is the size of the lexicon. Scaling the logit affects the probability assigned to each candidate word.

As the temperature decreases (T $\rightarrow$ 0), each term y_i/T becomes increasingly large for the largest y_i and increasingly small (negative) for all other values. Consequently, the exponential term exp(y_i/T) for the largest y_k rapidly dominates the denominator in the softmax calculation, causing $p_k \rightarrow 1$ for the token with the highest logit while all other probabilities $p_i \rightarrow 0$. In this limit, the output effectively approaches a "hard" argmax, where the model almost always selects the token with the highest predicted logit at each step, a behavior known as greedy decoding. Conversely, when T is larger, the resulting probability distribution becomes flatter and more uniform, allowing for greater diversity and randomness in the generated outputs. In summary, as $T \rightarrow 0$, softmax sampling becomes deterministic, consistently choosing the most likely token during generation. The value set for each is the parameter is summarized in Table 5.

Table 5. Tuning metrics.

Metrics	Value
Temperature	0
Output node limit	50
Fuzzy matching score	0.5

4.6. Formulation of Search Mechanisms

The search mechanism in GraphRAG consists of three layers: local search, global search, and hybrid search. In local search, the relevance score is calculated based on the cosine similarity between the query vector q and the embedding vector v_i of each node, using the following formula. The relevance score is calculated using the following formula.

$$s_i^{local} = \frac{q \cdot v_i}{\|q\| \|v_i\|} \tag{2}$$

This enables node selection based on semantic proximity to the query. In contrast, global search considers the network structure of the entire graph and defines a score s_i^{global} based on the semantic distance $d\ (q,\ n_i)$ between nodes using the following formula:

$$s_i^{global} = \frac{1}{1 + \ d\ (q,\ n_i)} \tag{3}$$

This distance metric is defined based on the shortest path length between nodes or semantic similarity, with shorter distances receiving higher scores. Furthermore, in hybrid search, both local and global scores are integrated to calculate the overall relevance score s_i^{hybrid}.

$$s_i^{hybrid} = \alpha s_i^{local} + (1-\alpha) s_i^{global}, \alpha \in [0,1] \tag{4}$$

The weighting coefficient α here is a parameter for adjusting the contribution of local semantic similarity and graph structural relevance and is adaptively set according to search objectives and data characteristics.

4.7. Community Detection

GraphRAG combines community detection to improve the accuracy of node relationships within the knowledge graph. Communities are identified by the Louvain method with the objective of maximizing modularity Q. Modularity Q is defined by the following formula:

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j) \tag{5}$$

Here, the adjacency matrix A_{ij} indicates the presence or weight of edges between node i and node j, and the degree k_i of node i is defined as the sum of all edges connected to that node (or the total edge weight if weighted). Furthermore, the total number of edges m in the entire graph is represented by half the sum of all elements in the adjacency matrix between all nodes. Each node i belongs to a community c_i, and the Kronecker delta $\delta(c_i, c_j)$ returns 1 if nodes i and j belong to the same community and 0 otherwise. By assigning nodes to communities to maximize this modularity Q, the Louvain method effectively detects densely connected clusters of nodes within the graph. Based on preliminary experiments, the minimum community size is set to 5 and the resolution parameter to 1.0, ensuring a balance between clustering granularity and interpretability.

4.8. Entity Duplication Integration

Entity merging was performed to consolidate identical or semantically equivalent nodes within the knowledge graph. The similarity of node pairs (v_i, v_j) is calculated as follows:

$$(v_i, v_j) \in C \quad \text{if} \quad sim(v_i, v_j) \geq \theta_{ER} \tag{6}$$

After integration, attribute a_k a merges attributes a_i and a_j of the original node using the integration function f:

$$a_k = f(a_i, a_j) \tag{7}$$

The edge weight $w_{k,u}$ after integration is calculated using the integration function g from the original node weights $w_{i,u}$ and $w_{j,u}$:

$$w_{k,u} = g(w_{i,u}, w_{j,u}) \tag{8}$$

After merging, perform similarity calculations again and iteratively process if any mergeable nodes exist:

$$\mathcal{V}' \to \mathcal{V}'' \quad \text{if} \quad \exists(v_p, v_q) \in \mathcal{V}', \quad sim(v_p, v_q) \geq \theta_{ER} \tag{9}$$

The final integrated node set $\mathcal{V}$ is obtained and reflected in the knowledge graph as unique entities without duplication.

4.9. Evaluation Metrics

To quantitatively assess the quality of responses generated by each strategy, six metrics were employed (Table 6).

Table 6. Evaluation metrics.

Metrics	Abbreviation	Description
Word Count	WC	Total number of words in a sentence.
Type Token Ratios	TTR	TTR assesses vocabulary diversity. A higher TTR indicates a more varied vocabulary.
Average Dependency Distance	ADD	ADD evaluates the distance between word dependencies in a sentence. A higher ADD suggests a more complex sentence structure and increased reading difficulty.
Flesch-Kincaid Grade Level	FKG	FKG is a readability measure that evaluates how challenging a text is to understand, based on the average length of its words and sentences. A higher score indicates greater difficulty.
BERT(P)	P	Precision (P) measures how well the words that appear in Sentence 1 are matched in Sentence 2.
BERT(R)	R	Recall (R) measures how many important words from sentence 2 appeared in sentence 1.
BERT(F)	F	F1 Score (F) measures the harmonic mean of Precision and Recall.

For BERTscore we adopt the three scores Precision (P), Recall (R), and F1 Score (F), which are computed as per the following quotations [25,26].

$$P = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{\hat{x}_i \in \hat{x}} x_i^\top \hat{x}_j \tag{10}$$

$$R = \frac{1}{|x|} \sum_{\hat{x}_i \in \hat{x}} \max_{\hat{x}_j \in \hat{x}} x_i^\top \hat{x}_j \tag{11}$$

$$F = 2\frac{P \cdot R}{P + R} \tag{12}$$

where x_i represent tokenized reference sentences, $x_i^\top$ are sequences of vectors generated by the embedding model, and $\hat{x}_j$ are tokenized candidates mapped to each of the vectors.

These metrics evaluated the relevance, coherence, and completeness of the answers generated by each approach.

5. Results

Prompts and Responses

We constructed a knowledge graph from the above processing (Figure A1). Based on the definitions presented in Section 4.2, we executed the prompt across all four models: (I) GPT alone, (II) local search, (III) global search, and (IV) hybrid search. The resulting outputs are shown in Tables 7–9.

The performance metrics (Table 10) further validated these findings, with Local Search (II) or Hybrid Search (IV) achieving the highest scores across all evaluation criteria, including relevance, coherence, and completeness.

Table 7. Output of "What new business models can you envision utilizing the Physical Internet?".

Model	Output
I	The Physical Internet enables new business models such as shared warehouses, flexible transport, and crowdsourced delivery. Companies can share resources without sharing sensitive data, helping build trust and collaboration. By using AI, IoT, and blockchain, logistics become smarter and more efficient. Smart city tech also helps improve urban delivery. These models reduce costs, save time, and support eco-friendly and flexible logistics systems that can adapt to changing needs.
II	The Physical Internet can drive innovations such as collaborative logistics platforms, on-demand delivery services, and smart supply chains that adapt to real-time demand. It also enables modular product design for recycling, dynamic pricing models based on data, and crowdsourced fulfillment to reduce delivery costs. These developments foster sustainability, efficiency, and customer-centric practices.
III	New business models leveraging the Physical Internet (PI) can include federated data spaces that enhance collaboration among logistics stakeholders, enabling synchromodal transport and dynamic freight planning. Innovations may involve decentralized logistics networks where companies share capabilities rather than sensitive data, fostering trust and flexibility. Additionally, integrating smart city technologies can optimize urban logistics, focusing on data sharing and stakeholder engagement to improve efficiency and sustainability. These models can create value by optimizing resource utilization and facilitating adaptive, financially viable logistics systems.
IV	New business models leveraging the Physical Internet (PI) can include federated data spaces that enhance collaboration among logistics stakeholders, enabling synchromodal transport and dynamic freight planning. Innovations may involve decentralized logistics networks where companies share capabilities rather than sensitive data, fostering trust and flexibility. Additionally, integrating smart city technologies can optimize urban logistics, focusing on data sharing and stakeholder engagement to improve efficiency and sustainability. These models can create value by optimizing resource utilization and developing new services that adapt to changing market demands and environmental challenges.

Table 8. Output of "What policy approaches should be taken to make the Physical Internet a reality in Global?".

Model	Output
I	The Physical Internet (PI) aims to create a global system, and for this, several policy strategies are needed. First, a legal framework should be set up, covering transport laws, antitrust rules, and contract responsibilities. It is important to make regulations the same at both the EU and global levels to help connect logistics. Cooperation between the public and private sectors, investing in smart infrastructure, and supporting new technologies are all crucial. Also, focusing on sustainability is important to make sure the Physical Internet helps with environmental goals.
II	To realize the Physical Internet, key policies include standardization of data formats and logistics processes for global interoperability. Infrastructure investment in IoT and automation will support efficient data exchange and goods movement. A robust regulatory framework is necessary to balance innovation with safety and sustainability. Encouraging collaboration, supporting R&D, and strengthening education and training programs will further accelerate progress. These policies will help achieve the vision of a globally connected Physical Internet.
III	The global realization of the Physical Internet requires a multifaceted policy approach. Legal harmonization across jurisdictions is essential to address transport laws, antitrust issues, and contractual complexities. Regulatory convergence and public–private collaboration are crucial for seamless implementation. Strategic investment in smart infrastructure will enhance interoperability and real-time data exchange. Innovation-driven research is key to advancing logistics technology. Ultimately, sustainability must be a core policy focus, ensuring ecological responsibility in logistics evolution.
IV	To implement the Physical Internet globally, several policy strategies are essential. First, a comprehensive legal framework must be developed, addressing transport law, antitrust law, and contractual obligations for cross-border implementation. Harmonizing regulations at the EU and global levels will reduce barriers to logistics integration. Stakeholder engagement, fostering collaboration between public and private sectors, is key to its successful development. Investments in smart infrastructure and support for innovation are crucial for technological integration. Additionally, sustainability must be prioritized to ensure the Physical Internet contributes to environmental goals.

Table 9. Output of "What areas of research are missing that focus on the Physical Internet?".

Model	Output
I	The Physical Internet (PI) aims to make logistics like the digital internet, but more research is needed on standardization, sustainability, security, and the use of drones and autonomous vehicles. There is a lack of studies on the risks, challenges, integration of networks, and business models for PI, as well as the optimization of hub locations and network resilience.
II	Research areas that are missing in the context of the Physical Internet include the development of standardized protocols for data exchange, integration of IoT technologies for real-time tracking and monitoring, sustainability practices in logistics, and the impact of automation and robotics on supply chain efficiency. Additionally, studies on the socio-economic implications and regulatory frameworks surrounding the Physical Internet are also lacking.
III	The areas of research that are missing in the context of the Physical Internet include the exploration of specific implementation barriers, the development of robust business models, and the investigation of stochasticity in hub location and network design. Additionally, there is a need for more studies on the resilience and robustness of hyperconnected networks, as well as frameworks defining hyperconnected networks in practical applications.
IV	The areas of research that are missing in the context of the Physical Internet include a comprehensive exploration of the strengths, risks, challenges, and potential barriers to implementation. Additionally, there is a lack of studies focusing on the integration of hyperconnected logistics networks, the operationalization of the Physical Internet using frameworks similar to the digital internet, and the development of robust business models to support its adoption. Further research is also needed on the stochasticity and resilience of hub location and network design in large-scale optimization problems.

Table 10. Performance metrics (Bold indicates highest score).

Prompt	**1**				**2**				**3**			
Model	**I**	**II**	**III**	**IV**	**I**	**II**	**III**	**IV**	**I**	**II**	**III**	**IV**
WC	81	61	92	**97**	**104**	82	81	100	70	67	70	**96**
TTR	0.741	**0.820**	0.793	0.784	0.711	0.744	**0.790**	0.700	0.671	**0.687**	0.686	0.625
ADD	2.782	**2.906**	2.898	2.763	**3.194**	2.533	2.580	**2.708**	3.500	**4.246**	3.265	3.232
FKG	11.0	17.1	20.6	**21.2**	13.5	17.2	**19.7**	17.1	17.0	**23.4**	20.5	19.2
BERT (P)	-	0.727	0.754	**0.765**	-	0.673	0.714	**0.836**	-	0.698	0.715	**0.768**
BERT (R)	-	0.750	0.717	**0.719**	-	0.679	0.735	**0.825**	-	0.697	**0.747**	0.721
BERT (F)	-	0.739	0.735	**0.741**	-	0.676	0.724	**0.831**	-	0.698	0.731	**0.745**

6. Discussion

6.1. Stakeholder-Specific Insights

A comparison of GPT-only model (I) and GraphRAG models (II and IV) is outlined in Table 11.

Table 11. Comparison of GPT-only model (I) and GraphRAG models (II and IV).

Feature	GPT-Only (I)	Local Search (II)	Hybrid Search (IV)
Specificity	Moderate	High	Very High
Novelty	Commonplace ideas	Emerging concepts	Advanced, integrative
Context	Low	Node-level	Node + community (local + global)
Actionability	Basic guidance	Operational insights	Strategic + system-wide recommendations
Stakeholder alignment	General	Contextualized	Deeply tailored

The tailored prompts revealed unique insights for each stakeholder group.

First, from industry perspective represented by prompt, "What new business models can you envision using the Physical Internet?". GPT-only (I) mentions general ideas like shared warehouses, smart logistics using AI/IoT, and crowdsourced delivery. Overall,

they are high-level and somewhat generic, lacking technical or systemic depth. While response from GraphRAG introduces specific mechanisms like collaborative platforms, modular design, real-time demand adaptation, dynamic pricing, connects business models to sustainability and customer-centricity in Local Search (II). The response from Hybrid Search (IV) suggests system-level innovations like federated data spaces and synchromodal transport, integration of urban smart city logistics and market/environmental adaptability. Overall, it highlights more holistic and grounded in technical context, reflecting insights from both node-level detail and community-level context in the graph. In other words, Local Search (II) and Hybrid Search (IV) each possess distinct characteristics. Local Search (II) excels at presenting realistic and actionable insights tailored to the specific needs of each stakeholder, making it suitable for short-term, practical recommendations. Conversely, Hybrid Search (IV) provides comprehensive insights from a long-term perspective that integrates technical, institutional, and societal aspects, making it suitable for optimizing the entire system and strategic decision-making. Therefore, it is crucial to explicitly state which search model is adopted based on the paper's purpose. Specifically, clearly indicating that Local Search (II) is chosen when prioritizing practicality and feasibility, and Hybrid Search (IV) when emphasizing comprehensive and strategic analysis, allows the reader to clearly understand the rationale behind the model selection.

Second, from government perspective, in response to "What policy approaches should be taken to make the Physical Internet a global reality?", GPT-only (I) lists common policy suggestions, such as legal frameworks, public–private cooperation, smart infrastructure, and sustainability focus, which are useful but conventional, particularly not tailored to PI's unique needs. Local Search (II) results offer detailed, actionable steps, such as standardization of data formats, infrastructure investments in automation/IoT, support for education and R&D, clearly addresses global interoperability and innovation readiness. Hybrid Search (IV) results propose structured and layered policy development, such as cross-border legal harmonization, incentives for sustainability, and smart infrastructure tied to stakeholder engagement. Overall, GraphRAG provides stronger emphasis on governance, integration, and ecosystem building—not just tech or funding.

Third, from academia perspective, the prompt "What research areas are missing that focus on the Physical Internet?" identified key gaps in the existing literature. GPT-only model (I) highlights broad areas such as standardization, sustainability, security, drones/AVs, which are good baselines but lack conceptual depth and specifics. Local Search (II) results identify concrete gaps, such as real-time IoT integration, automation impact studies, and socio-economic/regulatory frameworks which are more aligned with emerging research agendas. Hybrid Search (IV) results push boundaries by suggesting:

Cybersecurity frameworks for PI, hyperconnected network operationalization, stochastic and resilient hub-network design. Overall, they address methodological, structural, and practical gaps for real-world PI deployment.

These findings demonstrate GraphRAG's ability to generate actionable insights tailored to the needs of diverse stakeholders by leveraging its graph-based relational structure.

6.2. Implications on Knowledge Management

The study demonstrates GraphRAG's potential as a powerful tool for organizing and synthesizing fragmented knowledge in complex domains like PI. By structuring information into interconnected graphs and enabling flexible query mechanisms, GraphRAG bridges gaps between diverse data sources. This capability is particularly valuable for advancing research in emerging fields where knowledge is distributed across multiple disciplines and formats. GraphRAG's ability to tailor insights to specific stakeholder perspectives enhances its applicability in real-world decision-making contexts. Industry

practitioners can identify new business opportunities; policymakers can design informed strategies; and academics can uncover research gaps. All of these can be performed in a unified platform.

6.3. Managerial and Policy Implications

The findings of this study provide actionable guidance for a range of stakeholders. Industry practitioners can use GraphRAG to identify emerging business models and improve operational efficiency by systematically exploring technological and logistical trends. Policymakers can benefit from the robust tool that supports the development of harmonized, data-driven regulatory frameworks to drive PI adoption across different jurisdictions. For academic researchers, GraphRAG enables the identification of under-researched areas, thereby supporting more focused and impactful research initiatives.

By delivering tailored insights that address the specific needs of each stakeholder group, GraphRAG enables informed decision-making and promotes cross-sector collaboration.

6.4. Limitations and Future Work

This study possesses certain strengths, but it also has several limitations. First, it relies on static datasets. If source materials contain omissions or biases, their effects may be reflected in the output results. Second, as the knowledge graph expands, computational complexity may impact performance. Third, the effectiveness of fuzzy matching depends on the granularity and consistency of node labels. Furthermore, while GraphRAG tends to exhibit lower hallucination risk compared to standard LLM generation, this risk is not entirely eliminated. Even with source linking capabilities, the model may still generate inaccurate yet plausible descriptions, potentially undermining user trust in identified research gaps or findings.

To mitigate these risks, it is crucial to provide referable and transparent outputs, such as explicitly indicating which graph nodes or edges support each part of the response. This enables users to verify the basis of the model's answers and more effectively evaluate the reliability of the presented findings. While this research demonstrates the potential of GraphRAG for structuring and integrating knowledge, its inherent limitations must also be clarified. Unlike established methods such as Systematic Literature Review, Taxonomy, and Critical Review, GraphRAG primarily relies on a Retrieval-Augmented Generation (RAG) framework that extracts and links relevant information from databases and corpora. Consequently, research gap analysis using GraphRAG may lack the comprehensiveness and contextual depth provided by systematic approaches.

For example, even if GraphRAG detects a gap such as "lack of standard protocols," it cannot sufficiently evaluate whether subsequent research has already addressed it. This limitation stems from the design characteristics of RAG models, which are not inherently suited for critical integration of information or analysis of temporal evolution. Consequently, while GraphRAG is useful for initial mapping of research domains and hypothesis generation, it should ideally be combined with established literature review methods to comprehensively and validly analyze research gaps.

Future research could address these challenges by integrating real-time data streams, expanding multilingual and multimodal knowledge graphs, and improving graph structuring methods to enhance scalability and responsiveness. Implementing user-facing measures to strengthen transparency and source attribution would also be beneficial. Moreover, the findings of this study are preliminary, and empirical experiments will be necessary in the future to confirm the practical usefulness of the insights generated by GraphRAG.

7. Conclusions

This study provides preliminary evidence of the potential utility of GraphRAG as a domain-specific knowledge integration framework for the PI domain. By aggregating fragmented data from academic publications, conference proceedings, and strategic roadmaps into a unified knowledge graph, GraphRAG facilitates systematic exploration of trends, knowledge gaps, and actionable insights tailored to stakeholder needs. The framework leverages graph-based relational structures combined with natural language understanding to enhance information retrieval, particularly for complex or ambiguous queries, while maintaining flexibility for domain-specific applications.

Among the four retrieval strategies evaluated—GPT-only, local search, global search, and hybrid search—local and hybrid search methods appeared to perform relatively better in generating stakeholder-specific responses with greater depth, relevance, and clarity. The use of fuzzy matching also contributed to improved semantic retrieval, supporting the practical usability of the framework.

Unlike traditional systematic reviews or standard RAG-based analyses, this approach focuses on integrating full-text documents (not just abstracts), cross-modal sources (including government roadmaps and conference proceedings), and semantic relationship extraction, all mapped to domain-specific PI ontologies. This allows for granular, stakeholder-tailored queries and identification of nuanced trends and gaps that are difficult to capture with conventional literature reviews or simple keyword-based approaches.

While these findings are preliminary and exploratory, they suggest that GraphRAG can serve as a practical framework for domain-specific knowledge integration, supporting evidence-based decision-making and innovation in the PI context. More broadly, this study illustrates how AI-driven, domain-focused knowledge frameworks can facilitate systematic knowledge synthesis in other complex, interdisciplinary fields.

Author Contributions: Conceptualization, H.N., A.Y. and E.H.; methodology, H.N. and E.H.; software, H.N.; validation, H.N. and E.H.; data curation, H.N.; writing—original draft preparation, H.N. and E.H.; writing—review and editing, E.H. and A.Y.; visualization, H.N.; supervision, E.H.; funding acquisition, E.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JSPS KAKENHI Grant Numbers JP 23K04076.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLMs	Large language models
GraphRAG	Graph-based retrieval-augmented generation
NLP	Natural language processing
GPT	Generative pre-trained transformer
WC	Word count
TTR	Type token ratios
ADD	Average dependency distance
FKG	Flesch–Kincaid grade level
CS	Flesch–Kincaid grade level
AI	Cosine similarity
DX	Logistics digital transformation
PI	Physical Internet
GPS	Global positioning system

Appendix A. Constructed Knowledge Graph

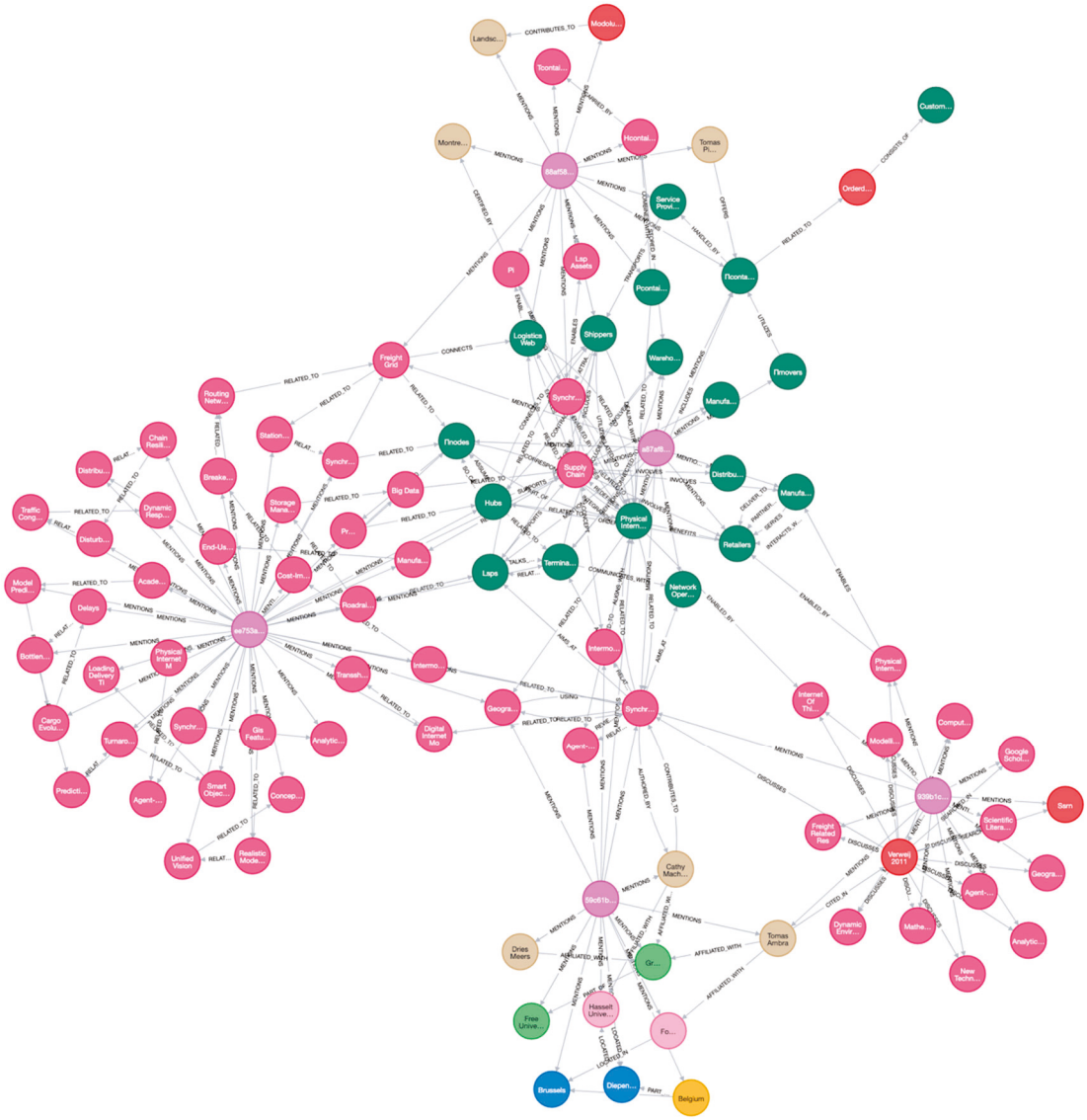


Figure A1. Knowledge graph.

Table A1 presents a statistical overview of the constructed knowledge graph. Note that the number of nodes displayed is limited to 100 to enhance the readability of the overall structure. This graph consists of 12,500 nodes and 24,800 relations extracted from 2835 documents. The average degree of 5.47 indicates moderate connectivity among concepts related to PI, confirming that fragmented knowledge has been semantically integrated. Furthermore, the modularity score of 0.72 calculated using the Leiden algorithm suggests that distinct topic-based communities, such as "Hub" and "Retailers," are clearly formed within the graph. This characteristic also contributes to improving the accuracy of subsequent local searches.

Table A1. Statistics of Knowledge Graph.

Metric	Value	Description
Graph Density	0.045	Ratio of actual edges to possible edges, indicating sparsity/density.
Average Degree	5.47	Average number of connections per node.
Document Nodes	2835	Source documents (Papers, Roadmaps).
Text Chunk Nodes	12,500	Segmented text blocks for granular processing.
Entity Nodes	24,800	Extracted domain concepts (e.g., "Logistics", "Hub").
Hierarchical Communities	17	Number of top-level clusters identified by Leiden algorithm.

References

1. Hirata, E.; Firdausiyah, N.; Kusumaningdyah, W. Handling Modular Containers in a Physical Internet Environment. *Marit. Econ. Logist.* **2024**, *27*, 449–475. [CrossRef]
2. Montreuil, B.; Meller, R.; Ballot, E. Towards a Physical Internet: The Impact on Logistics Facilities and Material Handling Systems Design and Innovation. In *Progress in Material Handling Research: 2010*; Material Handling Research at Georgia Southern Commons: Milwaukee, WI, USA, 2010.
3. Tallaki, S.; Abouelala, M.; Sekoun, A.K.; Mimouni, F.; Elbhiri, B. Insights into the Physical Internet: A Bibliometric Review. In *Industrial Innovations: New Technologies in Cities' Digital infrastructures*; El Bhiri, B., Merzouk, S., Assoul, S., Eds.; Sustainable Civil Infrastructures; Springer Nature: Cham, Switzerland, 2024; pp. 26–35, ISBN 978-3-031-70991-3.
4. Venkatadri, U.; Krishna, K.S.; Ülkü, M.A. On Physical Internet Logistics: Modeling the Impact of Consolidation on Transportation and Inventory Costs. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 1517–1527. [CrossRef]
5. Sternberg, H.; Norrman, A. The Physical Internet—Review, Analysis and Future Research Agenda. *Int. J. Phys. Distrib. Amp Logist. Manag.* **2017**, *47*, 736–762. [CrossRef]
6. Ballot, E.; Montreuil, B.; Zacharia, Z.G. Physical Internet: First Results and next Challenges. *J. Bus. Logist.* **2021**, *42*, 101–107. [CrossRef]
7. Aron, C.; Sgarbossa, F. The Physical Internet as an Approach for Resilient Logistics Practices: Literature Review and Future Research Avenues. *IFAC-Pap* **2023**, *56*, 11068–11075. [CrossRef]
8. Wu, H.; Li, M.; Yu, C.; Ouyang, Z.; Lai, K.; Zhao, Z.; Pan, S.; Wang, S.; Zhong, R.Y.; Kuo, Y.-H.; et al. Towards Cyber-Physical Internet: A Systematic Review, Fundamental Model and Future Perspectives. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *197*, 104051. [CrossRef]
9. Nguyen, T.; Duong, Q.H.; Nguyen, T.V.; Zhu, Y.; Zhou, L. Knowledge Mapping of Digital Twin and Physical Internet in Supply Chain Management: A Systematic Literature Review. *Int. J. Prod. Econ.* **2022**, *244*, 108381. [CrossRef]
10. Turing, A.M. Computing Machinery and Intelligence. In *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*; Epstein, R., Roberts, G., Beber, G., Eds.; Springer: Dordrecht, The Netherlands, 2009; pp. 23–65, ISBN 978-1-4020-6710-5.
11. Cascella, M.; Semeraro, F.; Montomoli, J.; Bellini, V.; Piazza, O.; Bignami, E. The Breakthrough of Large Language Models Release for Medical Applications: 1-Year Timeline and Perspectives. *J. Med. Syst.* **2024**, *48*, 22. [CrossRef] [PubMed]
12. McIntosh, T.R.; Susnjak, T.; Liu, T.; Watters, P.; Halgamuge, M.N. The Inadequacy of Reinforcement Learning From Human Feedback—Radicalizing Large Language Models via Semantic Vulnerabilities. *IEEE Trans. Cogn. Dev. Syst.* **2024**, *16*, 1561–1574. [CrossRef]
13. Oviedo-Trespalacios, O.; Peden, A.E.; Cole-Hunter, T.; Costantini, A.; Haghani, M.; Rod, J.E.; Kelly, S.; Torkamaan, H.; Tariq, A.; David Albert Newton, J.; et al. The Risks of Using ChatGPT to Obtain Common Safety-Related Information and Advice. *Saf. Sci.* **2023**, *167*, 106244. [CrossRef]
14. Rawte, V.; Sheth, A.; Das, A. A Survey of Hallucination in Large Foundation Models. *arXiv* **2023**, arXiv:2309.05922. [CrossRef]
15. Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; Huang, C. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv* **2024**, arXiv:2410.05779.
16. Sarmah, B.; Hall, B.; Rao, R.; Patel, S.; Pasquali, S.; Mehta, D. HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction. In Proceedings of the 5th ACM International Conference on AI in Finance, Brooklyn, NY, USA, 14–17 November 2024; pp. 608–616.
17. Li, Y.; Ko, H.; Ameri, F. Integrating Graph Retrieval-Augmented Generation With Large Language Models for Supplier Discovery. *J. Comput. Inf. Sci. Eng.* **2025**, *25*, 021010. [CrossRef]
18. Ojima, Y.; Sakaji, H.; Nakamura, T.; Sakata, H.; Seki, K.; Teshigawara, Y.; Yamashita, M.; Aoyama, K. Knowledge Management for Automobile Failure Analysis Using Graph RAG. In Proceedings of the 2024 IEEE International Conference on Big Data (BigData), Washington DC, USA, 15–18 December 2024; pp. 6624–6631.

19. Naganawa, H.; Hirata, E. Enhancing Policy Generation with GraphRAG and YouTube Data: A Logistics Case Study. *Electronics* **2025**, *14*, 1241. [CrossRef]
20. Markillie, P. The Physical Internet | 17 June 2006. Available online: https://www.economist.com/special-report/2006-06-17 (accessed on 28 November 2023).
21. ALICE Road Map to the Physical Internet. Available online: https://www.etp-logistics.eu/wp-content/uploads/2022/11/Roadmap-to-Physical-Intenet-Executive-Version_Final-web.pdf (accessed on 20 December 2024).
22. METI Physical Internet Roadmap. Available online: https://www.meti.go.jp/shingikai/mono_info_service/physical_internet/pdf/20220308_1.pdf (accessed on 20 December 2024).
23. Neo4j Aura. Available online: https://neo4j.com/product/auradb/ (accessed on 7 April 2025).
24. Sentence-Transformers/Paraphrase-MiniLM-L6-v2 · Hugging Face. Available online: https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2 (accessed on 19 February 2025).
25. Zhang, Q.; Chen, S.; Bei, Y.; Yuan, Z.; Zhou, H.; Hong, Z.; Dong, J.; Chen, H.; Chang, Y.; Huang, X. A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models. *arXiv* **2025**, arXiv:2501.13958.
26. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. *arXiv* **2019**, arXiv:1904.09675.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Voronoi-Induced Artifacts from Grid-to-Mesh Coupling and Bathymetry-Aware Meshes in Graph Neural Networks for Sea Surface Temperature Forecasting

Giovanny A. Cuervo-Londoño [1,*], José G. Reyes [2], Ángel Rodríguez-Santana [1] and Javier Sánchez [2]

[1] Instituto Universitario de Investigación en Acuicultura Sostenible y Ecosistemas Marinos (ECOAQUA), 35017 Las Palmas de Gran Canaria, Spain
[2] Centro de Tecnologías de la Imagen (CTIM), Instituto Universitario de Cibernética, Empresas y Sociedad (IUCES), 35017 Las Palmas de Gran Canaria, Spain; jsanchez@ulpgc.es (J.S.)
* Correspondence: giovanny.cuervo101@alu.ulpgc.es

Abstract: Accurate sea surface temperature (SST) forecasting in coastal upwelling systems requires predictive models capable of representing complex oceanic geometries. This work revisits grid-to-mesh coupling strategies in Graph Neural Networks (GNNs) and analyzes how mesh topology and connectivity influence prediction accuracy and artifact formation. This standard coupling process is a significant source of discretization errors and spurious numerical artifacts that compromise the final forecast's accuracy. Using daily Copernicus SST and 10 m wind reanalysis data from 2000 to 2020 over the Canary Islands and the Northwest African region, we evaluate four mesh configurations under varying grid-to-mesh connection densities. We analyze two structured meshes and propose two new unstructured meshes for which their nodes are distributed according to the bathymetry of the ocean region. The results show that forecast errors exhibit geometric patterns equivalent to order-k Voronoi tessellations generated by the k-nearest neighbor association rule. Bathymetry-aware meshes with $k = 3$ and $k = 4$ grid-to-mesh connections significantly reduce polygonal artifacts and improve long-term coherence, achieving up to 30% lower RMSE relative to structured baselines. These findings reveal that the underlying geometry, rather than node count alone, governs error propagation in autoregressive GNNs. The proposed analysis framework provides a clear understanding of the implications of grid-to-mesh connections and establishes a foundation for artifact-aware, geometry-adaptive learning in operational oceanography.

Keywords: sea surface temperature (SST); graph neural networks (GNNs); grid-to-mesh coupling; bathymetry-aware mesh; Voronoi partitions; artifact mitigation; operational oceanography

1. Introduction

The monitoring and forecasting of oceanographic variables have gained increasing importance over the last decade across various disciplines, including oceanography, climatology, sustainable marine resource management, maritime navigation, and natural disaster mitigation. Variables such as sea surface temperature (SST), salinity, currents, and sea level are central to understanding the dynamic state of the ocean and its interactions with terrestrial and marine ecosystems [1–3].

Growing concerns about climate change and the intensification of extreme events—such as hurricanes, marine heatwaves, and rising sea levels—have created

strong demand for fast, accurate, and computationally efficient predictive models to support decision-making in fisheries, maritime transport, marine conservation, and environmental policy.

Traditional physics-aware models provide a robust framework for simulating ocean dynamics, although limited by factors such as high computational costs, sensitivity to initial conditions, and the difficulty of representing complex marine environments on regular grids [4,5]. Deep learning methods have emerged as scalable alternatives for modeling oceanographic processes directly from observational data [6–8]. Among these, Graph Neural Networks (GNNs) are particularly well suited for geophysical applications, as they explicitly capture spatial relationships between observations through graph structures, offering flexibility for modeling localized and irregular phenomena [9,10].

Nevertheless, the performance of GNNs is strongly influenced by the interaction between the observational grid and the model's latent mesh [11]. Observational data, such as SST, are typically defined on structured spatial grids, and the mesh adopts a more flexible topological structure. The way these two entities are coupled defines how efficiently information flows during message passing and ultimately determines predictive accuracy. Empirical evidence [12,13] shows that inadequate grid-to-mesh coupling can generate systematic prediction artifacts, including polygonal patterns and discontinuities across spatial tiles, which not only increase errors but also undermine the physical consistency and reliability of forecasts over longer horizons.

This is particularly significant in strategically important regions such as the Canary Islands and the northwest African coast, where the upwelling process drives high biological productivity and sustains valuable fisheries. Accurate and timely SST forecasts in these regions are vital for sustainable resource management and for anticipating ecosystem responses to climate variability [14,15]. The recurrent appearance of localized artifacts in SST prediction highlights a structural weakness in current graph architectures, as errors tend to accumulate at tile boundaries and propagate through the prediction horizon.

To address this problem, we propose a systematic experimental framework that explores how mesh design and connectivity patterns influence error dynamics in SST forecasting. Conceptually, the mesh embeds the forecasting process, while the grid defines the physical observational lattice. By varying the number of connections per node, redistributing mesh nodes according to spatial complexity, and analyzing the resulting error maps, we aim to uncover reproducible design principles that reduce artifacts and improve forecast coherence. Our approach leverages high-resolution satellite products from Copernicus Marine and Climate services [16], supported by auxiliary information such as bathymetry and land–sea separation to ensure physically consistent data representation.

The aim of this work is thus twofold: first, to quantify how grid-to-mesh connectivity influences the geometry and severity of prediction artifacts; and second, to evaluate how strategic mesh reconfigurations can reduce errors and enhance the spatial coherence of forecasts. Preliminary findings [11] indicate that connectivity plays a crucial role, with approximately four connections per node mitigating artifact formation under the evaluated conditions, while naively increasing node count alone does not guarantee improvement. Instead, performance gains arise from node distributions in regions most susceptible to error. These insights open the path toward more robust grid-to-mesh association mechanisms.

We explore four different mesh configurations. Starting from a structured and quadrangular structure [17], we propose a simplified version removing crossing edges. Then, we derive two new unstructured mesh configurations based on a probability distribution of the bathymetry: The first approach relies on a random distribution inversely proportional to the depth of the sea, whereas the second redistributes nodes more regularly to avoid a high concentration near the coast.

Our results show that the optimal number of grid-to-mesh connections depends on the type of mesh employed, and densifying the node distribution according to the seabed's topology reduces artifacts and increases the accuracy of forecasts. The experiments demonstrate the better performance of bathymetry-based meshes with three and four connections, improving by 30% with respect to the structured mesh baselines.

Section 2 relates our work to the current state of the art, particularly in relation to forecasting with GNNs. In Section 3, we introduce the dataset and the geographical area of this study, as well as the different strategies employed for mesh generation. The results in Section 4 provide a detailed analysis of the influence of connections between the grid and the mesh in the generation of artifacts, as well as the importance of the number of nodes for improving accuracy. Finally, Sections 5 and 6 report the discussion of results and present the main findings of this work.

2. Related Work

Predicting oceanographic variables represents one of the most complex challenges in contemporary marine sciences, requiring the integration of multiple disciplines ranging from fluid physics and atmospheric dynamics to advanced computational techniques and artificial intelligence methods. The ocean, as a dynamic system, exhibits behaviors that span time scales from seconds to decades and spatial scales from meters to entire ocean basins, thus demanding diverse and complementary methodological approaches for its understanding and forecasting [18].

Traditionally, predictive oceanography has been grounded in physical ocean modeling, employing fluid mechanics equations to describe ocean dynamics through general circulation models, which have proven effective over decades. However, the exponential development of artificial intelligence techniques and the massive availability of oceanographic data have fostered the emergence of deep-learning-based ocean modeling as a complementary—and in some cases, alternative—approach to traditional methods [19].

Deep-learning-based methods promise to overcome some inherent limitations of physical models, such as the parameterization of subscale processes and high computational costs, using algorithms that can learn complex representations directly from data. The ability of these techniques to learn nonlinear representations from large volumes of data has driven their application in tasks such as SST prediction, reconstruction of incomplete fields, identification of dynamic structures, and detection of extreme events, among others [20].

Convolutional Neural Networks (CNNs) are particularly suitable for working with oceanographic data. Their convolutional layers can automatically extract relevant spatiotemporal patterns by leveraging the strong spatial and temporal correlations in marine data. Temporal Convolutional Networks [21] have been shown to capture multi-scale temporal dependencies in SST time series, and the combination of SST with auxiliary variables [22,23], such as sea surface salinity, sea height, and ocean heat, helps improve reconstruction and extend forecast horizons. To address spatial heterogeneity, the Multi-Scale Bayesian CNN [24] models uncertainty across scales, and 3D architectures like the 3D U-Net CNN [25] and Spatiotemporal Siamese CNN [26] capture complex spatiotemporal patterns, including regional variations and marine heatwaves. More advanced models [27,28] incorporate four-dimensional convolutions to represent vertical and horizontal ocean structures, enhancing the spatial resolution of satellite SST maps.

Recurrent Neural Networks (RNNs) have been widely applied in oceanographic modeling due to their ability to capture complex temporal dependencies associated with seasonal and interannual variability. LSTM [29] and GRU [30] models are commonly used for the univariate and multivariate forecasting of oceanic variables. Hybrid approaches combining RNNs with Convolutional Neural Networks (CNNs) exploit CNNs for spatial feature extraction and RNNs for temporal modeling. In particular, ConvLSTM networks [31] have shown strong performance in spatiotemporal forecasting tasks, such as in SST and chlorophyll prediction [32,33]. Further extensions [34–36], enhance temporal learning through graph structures, semantic decomposition, and multi-source data integration. Attention-based architectures [37] and hierarchical models [38] improve feature selection and long-term forecasting stability. Additionally, the combination of CNN-GRU with multilayer perceptrons [39] for multiscale learning and spatial correlation modeling demonstrates the growing sophistication of RNN-based frameworks in ocean prediction tasks.

GNNs are particularly well-suited for oceanographic modeling because they naturally represent irregular spatial relationships among geographically dispersed locations [12]. In such models, nodes typically correspond to specific geographic positions, while edges encode physical or functional connections, enabling both node and edge features to capture complex spatiotemporal dependencies. Recent GNN-based approaches for SST prediction have leveraged these capabilities through diverse architectural designs. For instance, GNN-OAM [40] integrates multiple adjacency matrices and attention mechanisms to model three-dimensional spatiotemporal interactions; GMNN [10] combines a graph encoder with memory units to capture long-term temporal dependencies; DYGOV [41] introduces a dynamic graph that evolves in response to oceanic and meteorological conditions; HiGRN [42] employs hierarchical node clustering with multi-level attention to integrate local and global spatial patterns, achieving improved global SST prediction performance.

GNNs are particularly effective for modeling complex geographic regions such as coastal zones, archipelagos, and areas influenced by dynamic ocean currents [12,13,17]. The design of the mesh structure plays a pivotal role in balancing model accuracy and computational efficiency. Global models such as GraphCast [43] employ multiresolution icosahedral meshes to represent the entire planet, albeit at the cost of a high node count. In contrast, local models like HI-LAM [44] use rectangular meshes to focus on specific regions, while SeaCast [17] adapts mesh contours to irregular coastlines, such as those of the Mediterranean, while maintaining a regular grid structure. However, regular meshes may introduce visual artifacts at lower resolutions [12,13], and simply increasing the number of nodes does not necessarily improve predictive precision and can even degrade performance due to computational inefficiencies.

Rather than assuming that denser or bathymetry-aligned meshes inherently improve accuracy [11], our findings indicate that the dominant factor is the association geometry between grid and mesh. Even topology-aware graphs impose order-k Voronoi partitions that act as semi-isolated predictors, leading to structured temporal drift along tile boundaries. Physical gradients like bathymetry are useful only when employed to reshape connectivity or redistribute nodes where per-tile RMSE slopes are steep, as they are not used as standalone prescriptions for mesh design.

3. Materials and Methods

In this section, we introduce the dataset and the area of study used in this work, related to the Canary Islands and the Northwestern African coast. Then, we explain the GNN architecture and the different mesh configurations tested in this work. Finally, we comment on the experiments and metrics used in the results.

3.1. Dataset

The experimental dataset is a spatiotemporal series of SST and wind fields over the Canary Islands region, derived from the Copernicus Marine Service Program and the European Centre for Medium-Range Weather Forecasts, respectively. Specifically, we use the SST reanalysis product [45], which provides daily observations at a spatial resolution of 0.05°. SST is a key physical variable for detecting upwelling processes and understanding coastal dynamics.

To complement SST, we incorporate wind data at 10 m above the surface, using the zonal (east–west) and meridional (north–south) components [46]. These variables provide valuable information for characterizing upwelling intensity along the African coastline. The temporal range covers 1 January 2000 to 31 December 2020, selected as the most recent interval available through Copernicus services.

The data were retrieved in NetCDF format. For each date-time d, the variables were organized into three-dimensional arrays ($A_d \in \mathbb{R}^{\phi \times \lambda \times f}$), where ϕ and λ represent the latitudinal and longitudinal coordinates of the 300 × 300 ocean grid cells, respectively, and f denotes the set of physical variables. Before storage, values were filtered using a land–sea mask, discarding land points. Missing values were set to zero, and dimensions were homogenized to ensure compatibility with GNN architectures.

The study area lies within the Iberia–Biscay–Ireland (IBI) oceanic region, as shown in Figure 1. Table 1 reports the spatial bounding box limits, covering coastal regions of Northwest Africa, including Capes Blanco, Jubi, and Ghir, along with the Canary and Madeira archipelagos.

Table 1. Geographic coordinates of the area of study.

Coordinates	Minimum (°)	Maximum (°)
Latitude	19.55	34.52
Longitude	−20.97	−5.98

Differentiating marine and terrestrial cells is essential for model training and inference. To this end, a binary land–sea mask was calculated, as shown in Figure 2. A bathymetry-based mask was also generated using NOAA's ETOPO model [47], providing depth information relevant for capturing nearshore variability. The bathymetric data were downscaled to match the grid resolution of the SST fields (see Figure 2). These masks supply physical context to the model and improve performance in regions characterized by strong topographic gradients.

3.2. Graph Neural Network and Mesh Design

The predictive framework employs an encoder–processor–decoder architecture based on interaction networks [48]. Message passing is used to exchange information across the latent mesh and the observational grid. This design aligns with recent geophysical models such as GraphCast [43], Hi-LAM [44], and SeaCast [17], which demonstrate the flexibility of bipartite grid-to-mesh graph $\mathcal{G} = \{\mathcal{V}^{\mathrm{G}}, \mathcal{V}^{\mathrm{M}}, \mathcal{E}^{\mathrm{M}}, \mathcal{E}^{\mathrm{G2M}}, \mathcal{E}^{\mathrm{M2G}}\}$ representations for data-driven climate forecasting. The model configuration used here isolates the role of mesh design and connectivity by holding hyperparameters and optimization choices constant across experiments.

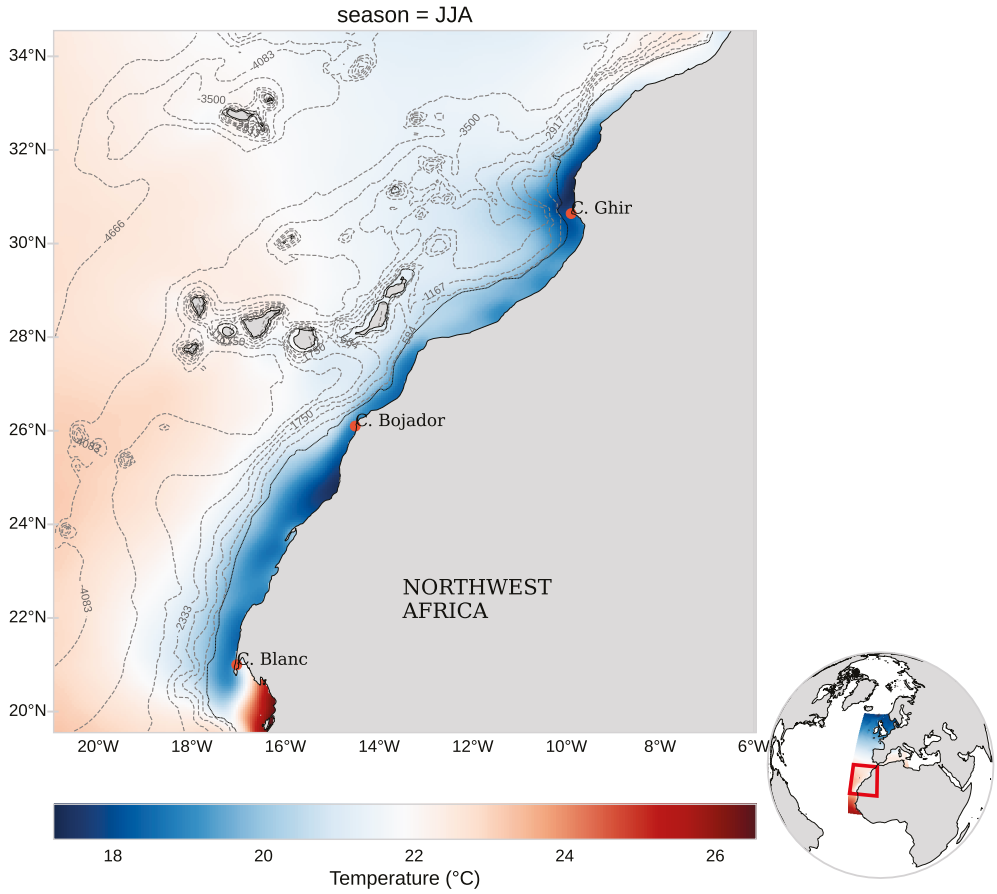


Figure 1. Summer (June–August) climatology of SST (°C) across the Northwest African region. The main figure illustrates the coastal temperature pattern. Dashed grey contour lines denote the bathymetry, highlighting seafloor features associated with upwelling processes. The small globe in the bottom-right corner shows the IBI (Iberian–Biscay–Ireland) region from the Copernicus dataset, with the study area outlined in red to indicate its position relative to the larger domain.

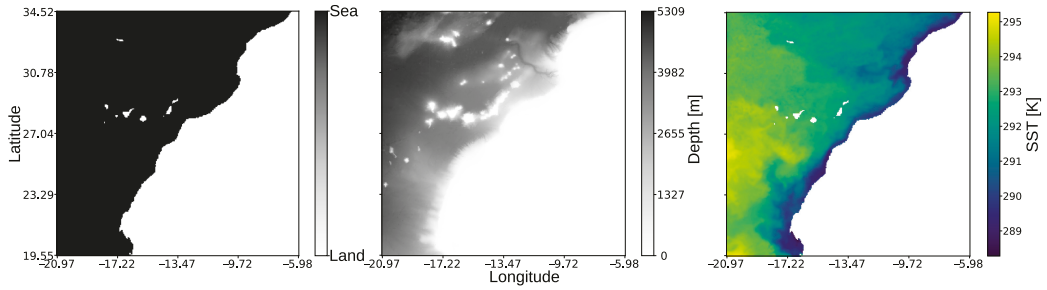


Figure 2. Spatial fields over the Canary Islands region. (**Left**): Land–sea mask distinguishing ocean (black) from land (white). (**Center**): Bathymetry (in meters). (**Right**): Sea surface temperature (SST, in Kelvin) on 1 January 2018.

Four mesh construction strategies were considered with the following configurations:

- Uniform Mesh with Crossing Edges (UC-mesh): Adapted from SeaCast, in this strategy, nodes lie on a regular orthogonal grid with up to eight connections, including diagonals. While this maximizes neighborhood density (Figure 3a), triangle intersections increase edge count and computational cost.
- Uniform Mesh without Crossing Edges (U-mesh): In this second strategy, the position of nodes is also orthogonal as before, and edges are obtained through a Delaunay triangulation. In this case, edges do not cross, avoiding diagonal intersections (see

Figure 3b). This reduces edge density while preserving spatial coverage. This configuration resembles the GraphCast model [43] in that edges do not cross, although it is a global atmospheric model that relies on an icosahedral mesh structure.

- Bathymetry-aware Mesh (B-mesh): In this case, node density is adapted to bathymetric variability, motivated by the greater bathymetry-driven SST gradients in coastal upwelling regions. Points are sampled according to probability distributions derived from normalized depth values. This improves spatial representation in coastal areas. Edges are also established using a Delaunay triangulation, as shown in Figure 3c. Based on the bathymetry ($\tilde{B}(x,y)$) shown in Figure 2, we normalize the values in $[0,1]$ using min–max normalization:

$$\tilde{B}(x,y) = \frac{B(x,y) - B_{\min}}{B_{\max} - B_{\min}}. \tag{1}$$

Nodes are placed based on the following probability function:

$$D(x,y) = \begin{cases} \dfrac{1}{\sqrt{\tilde{B}(x,y)}} & \text{if } \tilde{B}(x,y) > 0, \\ 0 & \text{if } \tilde{B}(x,y) = 0, \end{cases} \tag{2}$$

and then, the distribution $P(x,y) = D(x,y) / \sum_{x,y} D(x,y)$ is used to place nodes non-uniformly. This strategy adapts better to the coast geometry, although it tends to concentrate a large number of nodes near the coast, leaving few points to represent extensive deeper areas.

- FPS Bathymetry-Based Mesh (F-mesh): This strategy aims to solve the drawbacks of the previous one, maintaining a more homogeneous node separation while still relying on the bathymetry. It covers the whole region by distributing the nodes adaptively using the Farthest Point Sampling (FPS) algorithm [49] and a more balanced distribution. To achieve this, we use the following function:

$$D_i(x,y) = \begin{cases} \dfrac{1}{1 + \exp\left(\dfrac{\tilde{B}(x,y) - \tilde{B}_i}{\text{IQR}}\right)} & \text{if } \tilde{B}(x,y) > 0, \\ 0 & \text{si } \tilde{B}(x,y) = 0, \end{cases} \tag{3}$$

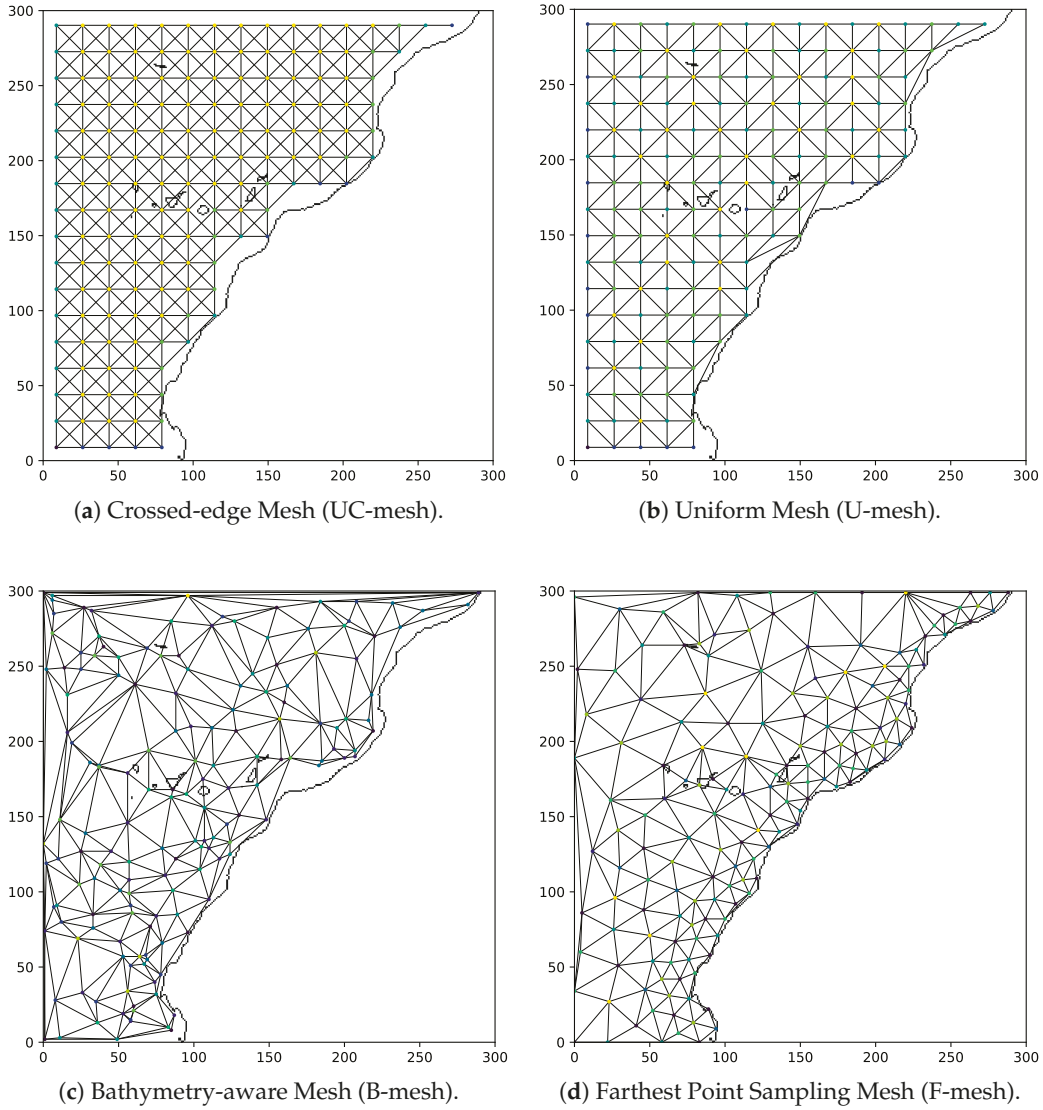
where $\tilde{B}_i$ represents the value corresponding to the i-th percentile of the bathymetry, and $\text{IQR} = Q_3 - Q_1$ denotes the interquartile range: that is, the difference between the third and first quartiles of the bathymetric distribution. We can then use the probability distribution $P_i(x,y) = D_i(x,y) / \sum_{x,y} D_i(x,y)$ to position the nodes in the mesh. This allows us to flexibly adjust the number of nodes in each area by varying the value of i, although we can also define a linear combination between two distributions as follows:

$$P(x,y) = \alpha \cdot P_i(x,y) + (1 - \alpha) \cdot P_j(x,y), \tag{4}$$

where $\alpha \in [0,1]$. If we choose $i = 25$ to give greater weight to the coastal zone and $j = 50$ to the deep ocean area, this relation allows us to flexibly balance the nodes between the coast and the ocean in a controlled manner. In our experiments, we used $\alpha = 0.7$. This mesh configuration is shown in Figure 3d.

All meshes are preprocessed by removing land nodes and equipped with an encoder/decoder scheme following the hierarchical design of [17,44]. Three edge types define the connectivity: First, bidirectional mesh-to-mesh edges ($e^{\text{M2M}} \in \mathcal{E}^{\text{M2M}}$) link each mesh node ($v^{\text{M}} \in \mathcal{V}^{\text{M}}$) to its spatial neighbors; second, grid-to-mesh edges ($e^{\text{G2M}} \in \mathcal{E}^{\text{G2M}}$)

connect every grid cell ($v^{\mathrm{G}} \in \mathcal{V}^{\mathrm{G}}$) to its k-nearest mesh nodes ($v^{\mathrm{M}} \in \mathcal{V}^{\mathrm{M}}$); third, mesh-to-grid edges ($e^{\mathrm{M2G}} \in \mathcal{E}^{\mathrm{M2G}}$) close the cycle by projecting information back to the grid. Unlike earlier implementations, the hyperparameter k explicitly enforces a fixed number of encoder/decoder connections ($e_k^{\mathrm{G2M/M2G}}$) per node (v_k^G, v^M), guaranteeing consistent coverage across the domain.



(**a**) Crossed-edge Mesh (UC-mesh). (**b**) Uniform Mesh (U-mesh).

(**c**) Bathymetry-aware Mesh (B-mesh). (**d**) Farthest Point Sampling Mesh (F-mesh).

Figure 3. Comparison of the four mesh-generation strategies used in the experiments. All meshes contain 159 nodes but differ in node distribution and edge construction: UC-mesh includes 1090 edges arranged in a crossed-edge configuration, increasing connectivity via diagonals; U-mesh consists of 848 non-crossing edges derived from a Delaunay triangulation; B-mesh contains 906 edges generated from a bathymetry-weighted probabilistic sampling given by (2); F-mesh uses 858 edges connecting sampled nodes along the coastline through an adaptive redistribution strategy calculated from the bathymetry as in (3).

3.3. Experimental Configuration

In this work, we designed two main experiments to assess the performance of the mesh configurations. On the one hand, we analyzed the influence of the connectivity between the grid and the mesh and its role in generating visible artifacts. On the other hand, we studied the influence of the number of nodes on the accuracy of the four strategies.

3.3.1. Experiment 1—Grid-to-Mesh Connectivity

The number $k \in \mathbb{N}^*$ of $e_k^{\mathrm{G2M/M2G}}$ connections per v^G node was varied between $k = 1$ and $k = 5$ across the four mesh types, yielding 20 different configurations. Each model was trained using the same number of epochs (150); the AdamW optimizer with $\lambda = 0.1$, $\beta_1 = 0.9$, and $\beta_2 = 0.95$; and a cosine-decay learning rate starting at 0.001. The loss function was the weighted mean squared error (WMSE), given by

$$\mathrm{WMSE} = \frac{1}{\sum_{i=1}^{N} m_i} \sum_{i=1}^{N} s\, m_i w_i\, (\hat{y}_i - y_i)^2, \tag{5}$$

where N is the number of nodes in the grid, $m_i \in \{0, 1\}$ is the binary land–sea mask defined in Section 3, and w_i is the latitude-based weight given by [50]

$$w_i = \frac{\cos(\mathrm{lat}_i)}{\frac{1}{N}\sum_j^N \cos(\mathrm{lat}_j)}, \tag{6}$$

which accounts for the varying size of the cells produced by the spherical coordinates, and s is the per-variable inverse variance of time differences. The ground truth value is given by y_i, and the prediction is given by $\hat{y}_i$ in each point i of the grid.

Table 2 shows the configuration of meshes used in this experiment, with the number of nodes and edges in each level. The number of nodes in the second layer of the F-mesh was increased because a node became isolated. This isolation was due to a combination of the morphology of the African coast and the operation of the farthest point sampling algorithm. By adding a single node, we achieved a fully connected mesh without compromising the comparison, as the total number of edges remains the same as in other configurations tested in this experiment.

Table 2. Node and edge counts for the mesh configurations used in the grid-to-mesh connectivity experiment. To ensure fair comparison across topologies, all four mesh types (U-mesh, UC-mesh, B-mesh, and F-mesh) were constructed with approximately equal node counts at each resolution level. The F-mesh has one more node in Level 2 to avoid an isolated node and to maintain a similar number of edges.

	Level 1				**Level 2**			
	U-Mesh	**UC-Mesh**	**B-Mesh**	**F-Mesh**	**U-Mesh**	**UC-Mesh**	**B-Mesh**	**F-Mesh**
Nodes	14	14	14	14	5	5	5	6
Edges	54	66	60	58	14	16	14	14

Finally, a factorial analysis contrasted the main effects of mesh configuration, connectivity, and node density, as well as their interactions, on the target metrics. Differences in means and variances across experiments, confidence intervals, and post hoc tests were used to derive practical design criteria for robust grid-to-mesh coupling in geospatial prediction tasks.

3.3.2. Experiment 2—Node Density

For each mesh configuration, the node density was increased following a geometric progression with ratio $\varphi \approx 1.618$, consistent with the refinement ratios suggested in the literature [51,52]. Fixed ratios simplify the control of spatial resolutions and error reduction, and using a golden-ratio increment offers a useful middle ground between typical refinement factors (≈1.5 to 2), giving meaningful resolution improvements without a sharp rise in computational cost. Alternative scaling schemes exist, but the golden ratio is favored for its balance between refinement detail and computational cost in adaptive physical simulations, where it reduces numerical discontinuities [53]. Similar advantages appear in

geometric subdivision methods using φ for smoother multiresolution transitions [54], and theoretical work shows that golden-scale hierarchies match scale-distributed processes like turbulence [55], supporting φ-scaling as suitable for multiscale modeling.

All experiments maintained consistent vertical levels and the same number of nodes across mesh types. Mesh characteristics, including nodes and edges at hierarchical levels, are summarized in Table 3.

Table 3. Node and edge counts per mesh configuration and hierarchical level. Each row corresponds to a target (Level 1 and Level 2) configuration, with node and edge counts reported for all four mesh types.

		Level 1				**Level 2**			
Config.	**Element**	**B-Mesh**	**F-Mesh**	**UC-Mesh**	**U-Mesh**	**B-Mesh**	**F-Mesh**	**UC-Mesh**	**U-Mesh**
14_9	Nodes	14	14	14	14	9	9	9	9
	Edges	60	48	66	54	32	24	36	30
20_9	Nodes	20	20	20	20	9	9	9	9
	Edges	94	82	100	82	32	24	36	30
34_20	Nodes	34	34	34	34	20	20	20	20
	Edges	168	160	194	160	94	82	100	82
52_27	Nodes	52	52	52	52	27	27	27	27
	Edges	274	248	310	256	128	120	146	122
159_78	Nodes	159	159	159	159	78	78	78	78
	Edges	906	858	1090	848	426	400	498	396

RMSE values are computed over the entire test set ad aggregated across the full date-time span (d) to obtain $\mathrm{RMSE}_{i,t}$, where i stands for the flattened latitude–longitude grid location, and t denotes the lead time.

4. Results

This section analyzes the effects of grid-to-mesh connectivity, node density, and mesh topology on forecast accuracy and artifact formation. Unless otherwise stated, all models were trained under identical optimization settings and evaluated over the same temporal horizons: lead times spanning from 1 to 15 days, with global RMSE and per-tile RMSE_l of spatial gradients as primary metrics. Spatial diagnostics include error maps and automatic artifact detection based on the geometry of grid-to-mesh associations.

4.1. Influence of Grid-to-Mesh Connectivity

Table 4 summarizes mesh performance across all connectivity settings. B-mesh and F-mesh achieved the lowest RMSE (0.24 at $k = 3$ and $k = 4$), which improved the results of the structured meshes by nearly 30%. Its refinement according to seabed morphology concentrates resolution in coastal and high-gradient regions. However, the performance of the B-mesh is more sensitive to connection count, with a greater variability than any other mesh, as shown in Table 5 with $std = 0.08$, which indicates a strong dependency on tuning.

F-mesh, on the other hand, delivered the most consistent performance across connection counts. Its best result, with an RMSE of 0.24 and $k = 4$ connections, is nearly as good as B-mesh, but with less variance. Therefore, F-mesh offers a strong compromise between accuracy and robustness. Structured meshes (U-mesh and UC-mesh) perform similarly, with U-mesh slightly outperforming UC-mesh at their respective optima. Crossing edges exhibit the lowest variability but consistently higher mean RMSE, indicating limited adaptability compared with triangular meshes.

Table 4. RMSE in Kelvin by mesh type and number of grid-to-mesh connections, k. The best value in each row is in bold.

Mesh Type	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
UC-mesh	**0.36**	0.42	0.39	0.42	0.4
U-mesh	0.42	**0.34**	0.44	0.37	0.37
B-mesh	0.48	0.38	**0.24**	0.34	0.36
F-mesh	0.38	0.42	0.34	**0.24**	0.35

When aggregated over all five connectivity levels, F-mesh achieved the best mean RMSE (0.35 ± 0.06), followed by B-mesh (0.36 ± 0.08), U-mesh (0.39 ± 0.04), and UC-mesh (0.40 ± 0.02). These statistics (Table 5) highlight not only accuracy but also variability across connection settings.

Table 5. Aggregate 15-lead-time RMSE statistics.

Mesh Type	Mean RMSE	Std. Dev.
F-mesh	**0.35**	0.06
B-mesh	0.36	0.08
U-mesh	0.39	0.04
UC-mesh	0.40	0.02

These findings align with qualitative error maps for the B-mesh configuration, as shown in Figure 4. Low-connectivity settings ($k = 1, 2$) accentuate polygonal discontinuities and tessellation artifacts, likely explaining the poor performance of B-mesh with $k = 1$, while intermediate connectivity ($k = 3, 4$) reduces such artifacts and improves spatial coherence. At $k = 5$, error maps show signs of plateauing or mild regression, likely due to over-coupling and amplified message-passing interference.

Figure 5 depicts the evolution of RMSE across forecast horizons for the B-mesh ($k = 3, 4$) and F-mesh ($k = 3, 4$) configurations. B-mesh with $k = 3$ and F-mesh with $k = 4$ consistently produce better results, while the errors of the F-mesh with $k = 3$ and B-mesh with $k = 4$ are higher. Differences are negligible in the first 4–5 days but increase progressively, reaching approximately 30% relative degradation by the 15th day, emphasizing the operational significance of topology–connectivity interactions.

Table 6 shows the training and inference times for each mesh and connection count. Increasing k increases the number of $e_k^{\mathrm{G2M/M2G}}$ and, therefore, memory and runtime. The computational cost increases linearly with the number of connections. During training, each additional edge adds on average between 6.6k and 6.8k seconds (≈1.8 h), corresponding to a relative increase of approximately 30% per edge compared to the $k = 1$ case. In inference, the impact is substantially lower: around 570 s per edge (≈10 min), equivalent to a ∼25% increase.

These results indicate that moderate connectivity (3–4 links per node) balances information flow and error control, while trimming excess edges reduces computational costs without performance loss. Bathymetry-based meshes, particularly F-mesh with mixed-sigmoid densification, yield smoother coastal errors, highlighting the importance of spatial adaptability. Forecast skills are mainly shaped by mesh topology after the first few days, and benefits become evident once resolution surpasses a critical threshold.

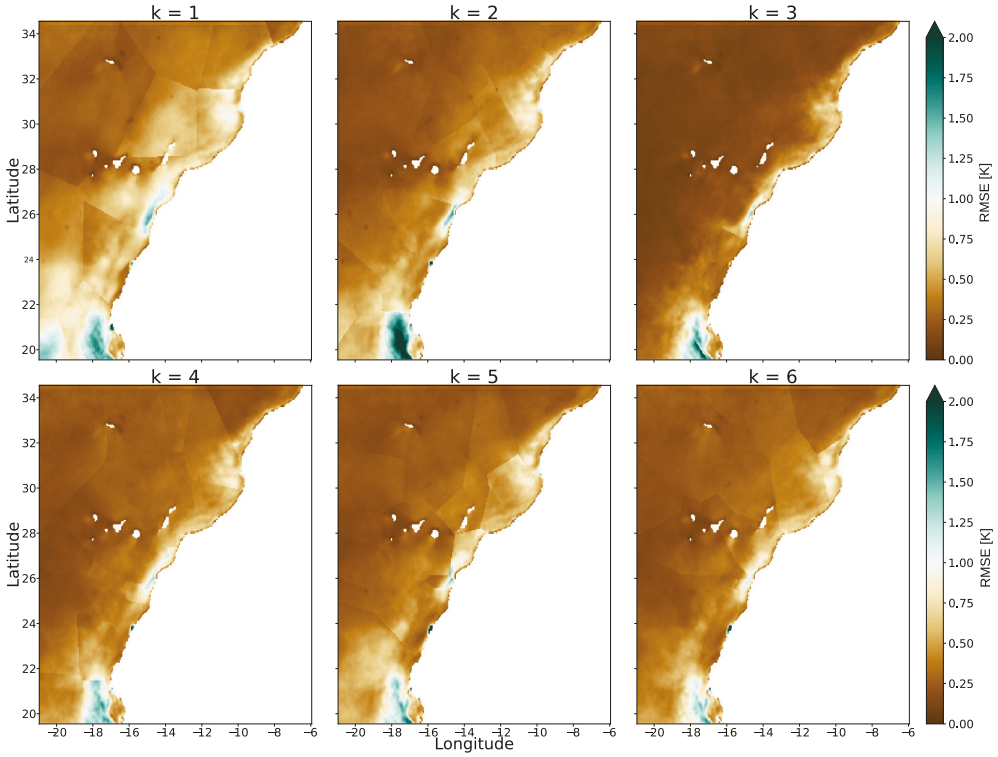


Figure 4. Spatial distribution of RMSE at 15-day lead time for the B-mesh configuration under varying connectivity levels ($k = 1$ to 6). Lower connectivity ($k = 1, 2$) produces remarkable polygonal discontinuities and tessellation artifacts, particularly along coastal headlands, whereas intermediate connectivity ($k = 3, 4$) improves spatial coherence and suppresses mesh-induced distortions. The results for $k = 6$ are included beyond the predefined range $k \leq 5$) to verify that performance does not improve for a larger number of connections, further supporting $k = 3$ as an optimal trade-off between coverage and over-coupling.

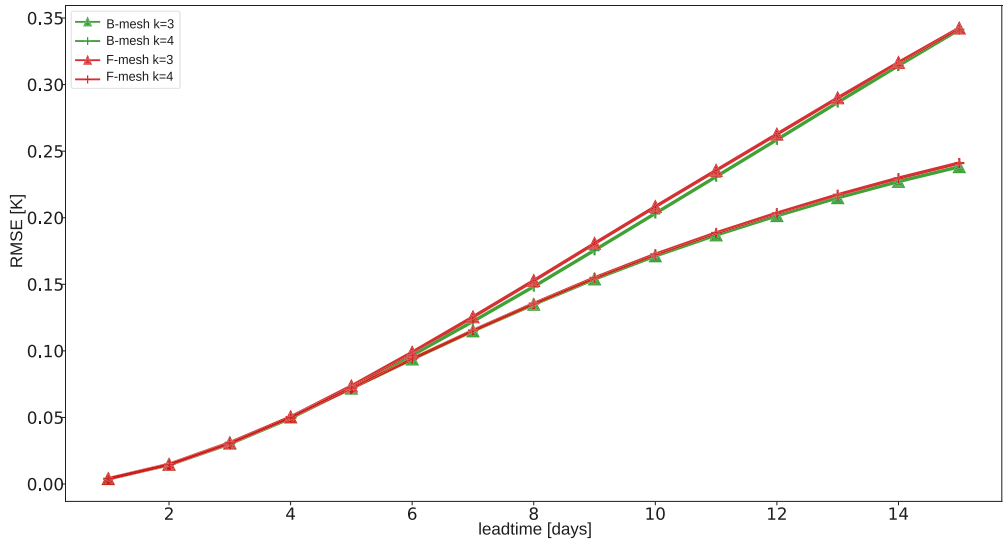


Figure 5. Average RMSE as a function of forecast lead time for the four best-performing topology–connectivity configurations: B-mesh ($k = 3, 4$) and F-mesh ($k = 3, 4$). Two distinct performance bands emerge beyond day 5, with B-mesh ($k = 3$) and F-mesh ($k = 4$) maintaining consistently lower error growth compared to B-mesh ($k = 4$) and F-mesh ($k = 3$). Error differences widen progressively toward day 15, highlighting the sensitivity of long-range forecasts to mesh topology and connectivity order.

Table 6. Training and inference runtime (seconds) across mesh configurations and k values, measured on a remote server with 8× Quadro RTX 4000 GPUs and 8 GB of memory.

k	Type	B-Mesh	F-Mesh	UC-Mesh	U-Mesh
1	Training	24,211	24,086	24,045	24,280
	Inference	1294	1353	1346	1350
2	Training	29,969	30,128	29,682	30,151
	Inference	1865	1917	1917	1918
3	Training	37,293	37,018	36,571	36,982
	Inference	2485	2536	2542	2537
4	Training	43,767	43,114	42,794	42,946
	Inference	2952	3005	3004	3002
5	Training	51,728	51,448	50,628	50,801
	Inference	3607	3660	3655	3653

4.2. Influence of Bathymetry and Regime Imbalance

The analysis of RMSE distribution across bathymetric ranges, visualized through box plots and sample size histograms (Figure 6), reveals a critical physical regime imbalance. The upper histogram highlights a skewed data distribution, with the coastal domain (<500 m) representing a significantly underrepresented minority compared to the open ocean. This scarcity correlates directly with model instability: The box plots in the shallow regime exhibit the largest interquartile ranges and whisker extensions, indicating maximum error variance and uncertainty. In contrast, the deep ocean regime (>1000 m) is characterized by compact boxes and stable medians, reflecting the GNN's robust performance in the dominant, physically homogeneous class. These results suggest that the network's optimization is biased towards the abundant deep-water samples, failing to generalize to the complex, data-scarce coastal dynamics where boundary effects and local variability prevail.

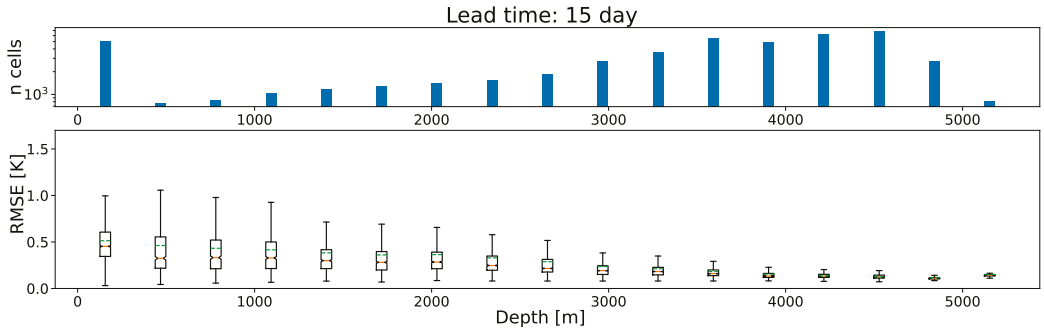


Figure 6. Dependence of predictive performance on ocean depth for B-mesh ($k = 3$) at 15-day forecasts. (**Top panel**): Histogram showing the frequency of grid cells across bathymetric ranges, illustrating the skewed distribution of data towards deep-water regions. (**Bottom panel**): Box plots of RMSE (K) versus depth (m). The plot reveals an inverse relationship between bathymetry and error variability: Coastal and continental shelf regions (0–1000 m) are characterized by high uncertainty and large error spread, whereas the abyssal plain (>3000 m) demonstrates robust model performance with compact error distributions.

4.3. Artifact Formation and Order-k Voronoi Effects

The spatial organization of the prediction error reveals a strong geometric imprint induced by the grid-to-mesh association. For the lowest connectivity ($k = 1$), the resulting partition closely resembles a first-order Voronoi diagram: Each mesh node defines a convex region, and the associated boundaries emerge where the Euclidean distance of two nodes is equal. The discontinuities in the RMSE field align with these borders, forming sharp, polygonal divisions across the domain (Figure 7). In this regime, tile boundaries are

simple and coherent, and the geometry is dominated by the distance-based isolation of individual nodes.

This observed geometric structure is not incidental but arises because the underlying grid-to-mesh coupling mechanism is mathematically equivalent to generating an order-k Voronoi diagram [56]. The formulation begins by defining the total number of mesh nodes as $n = |\mathcal{V}^{\mathrm{M}}|$. The association mechanism, often implemented as the spatial distance $d(\cdot,\cdot)$ (e.g., KD-tree search), identifies a generator subset $P^k(v^{\mathrm{G}}) \subset \mathcal{V}^{\mathrm{M}}$ for every grid node $v^{\mathrm{G}} \in \mathcal{V}^{\mathrm{G}}$. This subset consists of the $k \in \mathbb{N}^*$ mesh nodes closest to v^{G}. Crucially, k represents the fixed number of grid-to-mesh/mesh-to-grid connections ($e_k^{\mathrm{G2M/M2G}}$) per node. The set of all possible unique generator subsets is $\mathcal{A}^k$, where $l = |\mathcal{A}^k| \leq \binom{n}{k}$. Each distinct subset $P_i^k \in \mathcal{A}^k$ is explicitly defined as containing k mesh nodes, $P_i^k = \{v_1^{\mathrm{M}}, \ldots, v_k^{\mathrm{M}}\}$.

Each subset P_i^k defines a unique region known as an Order-k Voronoi Polygon, VP_i (Equation (7)). This specific polygon, VP_i, represents one of the distinct polygonal error regions (artifacts) observed in the prediction maps. It encompasses all grid nodes v^{G} that satisfy the following geometrical constraint:

$$\mathrm{VP}_i = \left\{ v^{\mathrm{G}} \in \mathcal{V}^{\mathrm{G}} \mid \max_{v_a^{\mathrm{M}} \in P_i^k} \left(d(v^{\mathrm{G}}, v_a^{\mathrm{M}}) \right) \leq \min_{v_b^{\mathrm{M}} \in \mathcal{V}^{\mathrm{M}} \setminus P_i^k} \left(d(v^{\mathrm{G}}, v_b^{\mathrm{M}}) \right) \right\} \tag{7}$$

The full set of these polygons, $\mathcal{VD}_k = \{\mathrm{VP}_1, \ldots, \mathrm{VP}_l\}$, is the induced order-$k$ Voronoi diagram. This complete tessellation, which reflects deterministic changes in neighbor sets, constitutes the spatial organization of the prediction error field. In the autoregressive GNN, this partitioning ensures that each Voronoi tile VP_i behaves as a semi-independent predictor, causing errors to accumulate with distinct, tile-specific growth slopes over long lead times.

This equivalence (Equation (8)) holds because the membership of a grid node (v^{G}) in an Order-k Voronoi polygon (VP_i) is entirely determined by its set of k nearest neighbors v^{M}. This relationship is formally proven by biconditional equivalence:

$$v^{\mathrm{G}} \in \mathrm{VP}_i \iff P^k(v^{\mathrm{G}}) = P_i^k \tag{8}$$

where $P^k(v^{\mathrm{G}})$ is the function that returns the k-nearest neighbor set (P_i^k) of v^{G}. Since each polygon (VP_i) is uniquely defined by a single k-element generator subset ($P_i^k \subset \mathcal{V}^{\mathrm{M}}$), the double inclusion is fulfilled: When the neighbor set of v^{G} coincides with this generator subset, the point satisfies the defining condition of the corresponding order-k Voronoi polygon; when it differs, it necessarily lies outside its boundaries. Thus, the biconditional relationship arises directly from the definition of the order-k diagram, where region boundaries occur precisely at locations v^{G} where its associated P_i^k changes.

As connectivity increases, the geometry of the induced partitions undergoes a qualitative shift. Instead of single-node dominance, each region is defined by overlapping neighborhoods of multiple mesh nodes. The resulting tiles lose convexity, become irregular, and exhibit fragmentation or elongation in certain areas. This increase in geometric complexity is spatially heterogeneous: Some zones retain simple structures, while others subdivide into non-convex or highly anisotropic shapes. The average tile area decreases with higher k, but this refinement does not translate into smoother error fields. Instead, the boundaries persist and frequently intensify, especially at vertices where three or more tiles converge, which act as localized amplification points for error growth under autoregressive prediction.

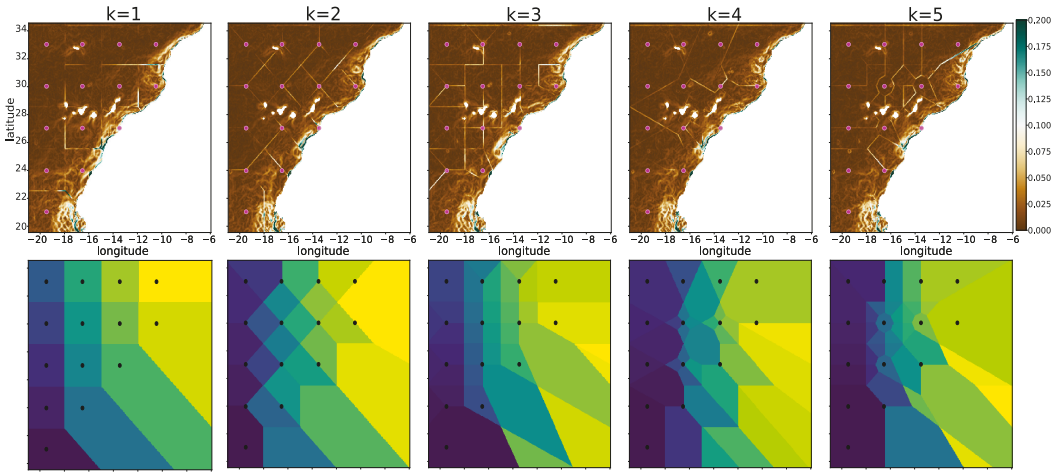


Figure 7. Spatial RMSE gradients at 15-day lead time highlight deterministic tiling artifacts imposed by the grid-to-mesh connectivity (U-mesh). Although the polygonal discontinuities are already present in the RMSE maps, the $|\nabla \text{RMSE}|$ representation makes the tile boundaries sharply visible. Magenta color points denote mesh node positions over the spatial domain. Top panels show the empirical gradient fields for different k settings, while bottom panels display analytically generated order-k Voronoi partitions using the same neighbor rule, reproducing the same geometry. This confirms that at longer lead times, the error field is governed by the tessellation induced by the connectivity mechanism rather than stochastic model behavior.

To confirm the geometric origin of these patterns, the connectivity mechanism was replicated using a synthetic mesh placed over a 300×300 domain (Figure 8). Each grid point was assigned its k-nearest neighbors via the same KD-tree-based search used in the encoder–decoder forecasting framework. The resulting partitions reproduced the theoretical structure of order-k Voronoi diagrams described in the analytical literature [57], matching the reference formulation precisely. Applying the same procedure to the U-mesh configuration from $k = 1$ to 5 yielded a tessellation with boundaries replicating those artifacts, validating that the spatial organization of the RMSE gradients is a direct consequence of the k-NN association (Figure 7). The discontinuities in the error are therefore not incidental: they reflect deterministic changes in neighbor sets across partition interfaces.

Coastal boundaries truncate tiles and remove candidate neighbors, creating smaller and more irregular cells than in offshore regions. Euclidean distances in latitude–longitude space further introduce directional bias, particularly along the north–south axis, distorting partition geometries. These geometric distortions interact with strong coastal SST gradients and localized heterogeneity, amplifying error accumulation. To systematically capture these effects, we introduce Error Analysis by Spatial Tessellation in Appendix A, a diagnostic framework that partitions the domain into tiles and tracks the evolution of forecast errors within each cell. Histograms of per-tile RMSE reveal heavy-tailed behavior near the coast (Figure A2), while offshore tiles display more symmetric distributions. The tile-averaged RMSE grows approximately linearly with lead time (Figure A3), but growth rates vary sharply across space, with the steepest slopes concentrated along coastal bands.

Increasing k does not eliminate these discontinuities. Although higher connectivity reduces individual tile size (Figure 7), it simultaneously increases fragmentation, irregularity, and the frequency of intersections. The transitions between neighbor sets remain discrete, preserving the geometric basis of the discontinuities. RMSE gradients continue to align with the tile boundaries, and vertices formed by multi-tile intersections become preferential sites for accelerated error propagation. The persistence of these structures

indicates that artifacts arise from the intrinsic geometry of the k-NN mechanism rather than from insufficient neighbor node count or model performance.

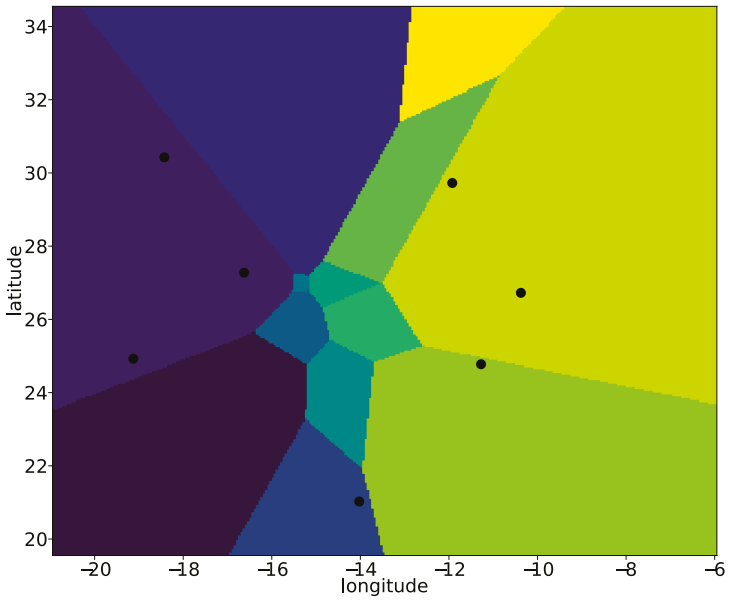


Figure 8. Reproduction of the order-3 Voronoi diagram from [57], generated using the same site configuration. In our framework, applying a grid-to-mesh association with $k = 3$ nearest neighbors via KD-tree yields the same partition geometry, demonstrating that the connectivity mechanism is mathematically equivalent to an order-3 Voronoi construction. This validates that the polygonal artifacts observed in the $|\nabla \text{RMSE}|$ fields arise directly from the deterministic tessellation induced by the k-NN association rule. Distinct colors represent different Voronoi regions, each defined by a unique set of k-nearest neighbors.

The spatial structure of the RMSE is dictated by the grid-to-mesh association, where tile geometry depends on connectivity and node placement. In autoregressive settings, each tile acts as a semi-independent predictor with distinct error dynamics (Figure A3), producing spatial–temporal heterogeneity. Artifact formation arises from the equivalence between connectivity rules and a Voronoi partition, implying that mitigation requires altering the association geometry rather than merely increasing connectivity.

4.4. Influence of the Number of Mesh Nodes

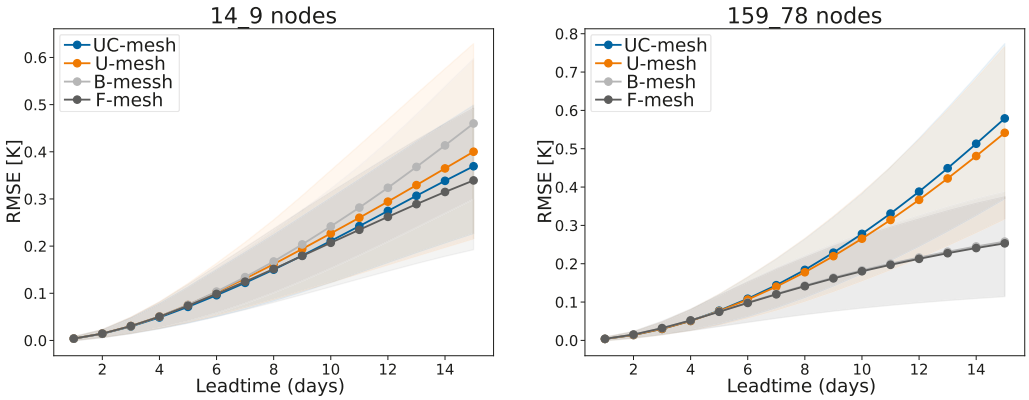
Across the four mesh configurations, RMSE trajectories remain nearly identical during the first four to five lead-time days, regardless of node count, as can be observed in Figure 9a. At these short horizons, forecast accuracy is primarily governed by initial observational states and the intrinsic predictive skill of the model, leaving mesh topology with negligible influence.

To rigorously quantify forecast uncertainty and validate performance differences, the posterior distribution of the RMSE was modeled using Bayesian Neural Fields [58], generating a probabilistic ensemble of $N = 240$ samples. We applied a mean bias correction ("Error Dressing"; [59]) and evaluated comparative performance using the distribution of paired differences ($d = RMSE_A - RMSE_B$), following [60] to account for temporal correlation. Significance was determined via a Zero-Inclusion Criterion on the 66% credibility intervals, as described by [61].

Applying this framework to the high-density node configuration (Figure 9b) reveals a statistically significant divergence from day six onward. While structured meshes (U-mesh and UC-mesh) show a steep increase in error, F-mesh and B-mesh sustain lower error growth with paired difference intervals that consistently exclude zero. This confirms the

existence of a temporal activation window: beyond day 6, error accumulation due to autoregressive structural artifacts is significantly mitigated by the adaptive geometry of unstructured meshes.

Conversely, when node counts are reduced, these statistical distinctions vanish across the entire prediction horizon (Figure 9a). The 66% credibility intervals for the differences largely overlap with zero, indicating that insufficient spatial resolution suppresses topology effects. These results identify a spatial resolution threshold: below a critical node density (e.g., 34 or 20 nodes), representational capacity becomes the dominant limiting factor, rendering the choice between structured and unstructured designs statistically negligible.



(**a**) Low-resolution mesh (14_09) configurations. (**b**) High-resolution mesh (159_78) configurations.

Figure 9. Forecast RMSE trajectories with associated uncertainty across prediction horizons for different mesh configurations under two levels of spatial resolution. Solid lines represent the RMSE mean (dressed RMSE), while shaded regions denote the 66% confidence intervals. (**a**) When the number of mesh nodes is low (14_9), all four configurations exhibit nearly parallel error evolution, with only minor divergence emerging after day 5. The substantial overlap of the confidence intervals confirms that the minor divergence observed after day 5 is not statistically significant. The limited spatial resolution constrains information propagation, suppressing the advantages of unstructured meshes. (**b**) At higher node density (159_78), the separation beyond day 5 becomes much more pronounced. Unstructured meshes (F-mesh and B-mesh) sustain lower error growth, with confidence bounds distinct from the structured baselines, indicating superior long-range predictive skill once spatial resolution is sufficient to exploit their adaptive placement.

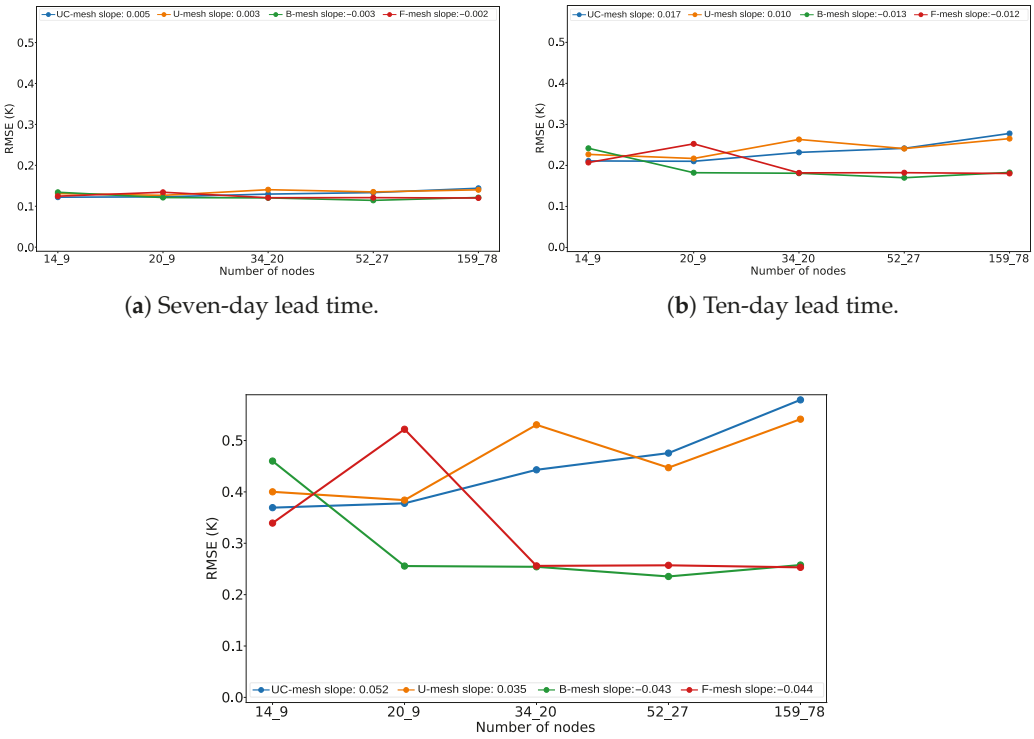
A secondary observation relates to the role of diagonal crossing edges in the UC-mesh. Configurations without crossing edges (U-mesh) occasionally show slightly more stable trajectories than those with intersecting connections (UC-mesh), suggesting that excessive edge proliferation may compound long-range error growth. However, this effect remains secondary to the broader distinction between structured and unstructured designs.

At the seventh-day lead time, all configurations show marginal differences (Figure 10a), with structured meshes sometimes presenting slightly lower RMSE than unstructured ones. The error curves remain nearly flat across resolutions, reinforcing the dominance of initial conditions at very short horizons.

By the tenth day lead time, a transitional regime emerges. Structured configurations begin to show a gradual increase in RMSE with higher node counts (Figure 10b), while the unstructured meshes remain stable or even improve modestly, particularly in the bathymetric layout. This suggests that once the constraint of the initial state weakens, topology-sensitive designs can better capture underlying SST variability.

At the fifteen-day horizon (Figure 10c), the divergence becomes pronounced. Structured meshes degrade substantially with increasing node count—especially in crossing-edge cases—whereas unstructured designs either maintain or reduce RMSE levels as

resolution increases. This inversion underscores that unstructured strategies not only accommodate spatial heterogeneity more effectively but also scale more robustly with forecast length.



(**a**) Seven-day lead time.

(**b**) Ten-day lead time.

(**c**) Fifteen-day lead time.

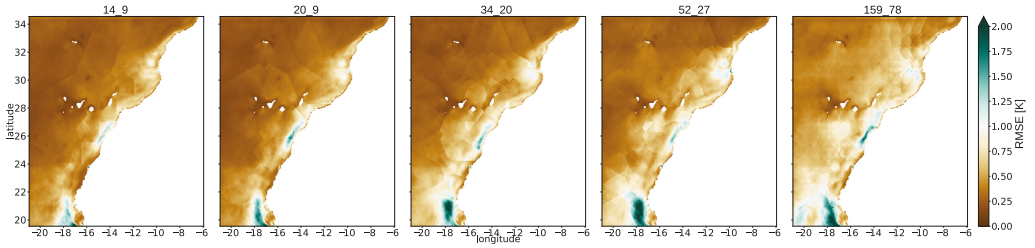
Figure 10. Comparison of RMSEs across node densities at different lead times. While all configurations behave similarly at seven lead times, structured meshes progressively degrade with higher node counts at ten and fifteen lead times. In contrast, unstructured layouts exhibit more stable or improving error performance as resolution increases.

Increasing node count alone does not guarantee improved forecast accuracy. At higher spatial resolutions, differences between mesh families become apparent only beyond day five. Here, unstructured meshes consistently outperform structured ones (Figure 10c), demonstrating that the spatial organization and connectivity—not raw node count—determine whether added resolution yields meaningful gains.

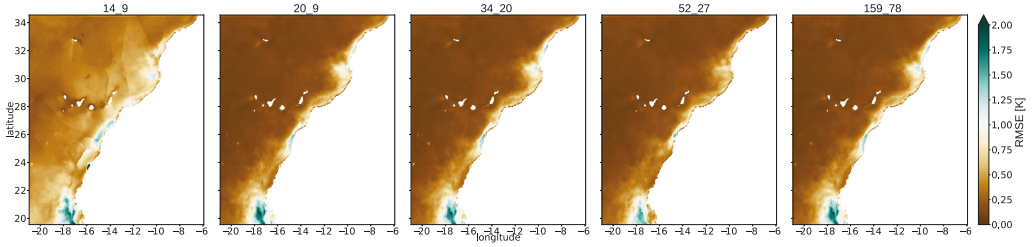
Notably, the B-mesh is highly effective at higher resolutions but performs comparatively worse at low node densities, indicating that adaptive placement requires a minimum number of nodes to operate effectively. F-mesh exhibits an intermediate behavior but similarly loses advantage when spatial capacity is limited.

Spatial analyses of squared-error fields at day 15 reveal clear contrasts among mesh families as resolution increases. In structured meshes, artifacts remain visible even at higher node counts (Figure 11a), although their footprint shrinks. Coastal regions consistently emerge as hotspots of error due to the rigid and uniform connectivity of these grids, which limits their ability to represent strong nearshore gradients.

Unstructured configurations exhibit smoother and more coherent error fields (Figure 11b), with far fewer discretization artifacts. The B-mesh shows the clearest advantage: Errors near the coast are reduced, and offshore propagation is constrained more effectively than in structured designs. Because node placement reflects underlying morphology, transitions between coastal and open-ocean regions are better resolved, attenuating artificial discontinuities.



(**a**) UC-mesh at 15-day lead time.



(**b**) B-mesh at 15-day lead time.

Figure 11. Spatial distribution of RMSE at 15-day lead time across increasing node densities for UC-mesh and B-mesh configurations. Structured grids exhibit persistent polygonal tessellation artifacts and concentrated coastal errors even at higher resolutions, reflecting limited adaptability to sharp shoreline gradients. In contrast, the unstructured B-mesh produces smoother and more coherent error fields, notably attenuating nearshore RMSE and constraining offshore error propagation due to morphology-aware node placement.

F-mesh shows intermediate behavior. They benefit from increased resolution and lack of rigid cell imprinting, but they do not suppress coastal error growth as consistently as the bathymetric layout. Across lead times, unstructured meshes modulate not only the magnitude but also the spatial evolution of RMSE. At higher node densities—particularly in the B-mesh configuration—error growth remains more localized and physically coherent, aligning with the statistical trends observed in the RMSE curves.

5. Discussion

A decisive step in interpreting these results is the geometric diagnosis of artifacts. Geometric discretization schemes, such as Voronoi diagrams, can either introduce or reveal heterogeneity. While such structures may be advantageous in physics-based numerical simulations [62], in autoregressive GNN-based models, they can become an unintended source of structured error. The grid-to-mesh association induces partitions that are algebraically equivalent to order-k Voronoi diagrams [57]; tile boundaries in these diagrams align with the polygonal error patterns observed in the RMSE maps. This explains the persistence of artifacts in autoregressive settings: Each tile behaves as a partially isolated predictor with its own error trajectory (Figure A3), producing approximately linear per-tile RMSE slopes that diverge over lead time. Hence, the core issue is not only error magnitude but the spatial–temporal heterogeneity of error accumulation determined by association geometry.

While geometric analogies with traditional discretization methods such as the finite element method are conceptually relevant—both involve mesh-induced partitions that shape error patterns—the present work does not aim to establish a formal numerical equivalence or perform a horizontal comparison with FEM. Our focus is instead on diagnosing artifact formation within GNN-based grid–mesh associations.

Beyond the geometric artifacts discussed above, a secondary source of error arises from the unequal distribution of training data across depth regimes. Regardless of the mesh topology, the error magnitude intensifies significantly in nearshore zones. As shown

in Figure 6, the error profile reveals a bottleneck in shallow waters that is distinct from the tessellation patterns: The model exhibits its highest uncertainty in these regions due to a severe class imbalance between the dominant deep-ocean samples and the narrow coastal margins. This class imbalance is consistent with the inherent morphology of the ocean floor. The spatial extent of abyssal plains (typically ranging from 3000 to 6000 m) vastly exceeds that of continental shelves (0–200 m) and slopes, which exist only as narrow transitional margins bordering the landmasses. Consequently, the dominance of deep-water samples is not a sampling artifact, but a direct reflection of the natural bathymetric distribution within the study area.

Some studies suggest a direct trade-off between accuracy and node count [63], assuming that denser meshes typically lead to lower error, reinforcing the idea that more nodes yield better accuracy. However, our results bring into question this perspective by clarifying why indiscriminate increases in k connections or node count are insufficient. While unstructured meshes (F-mesh and B-mesh) coupled with intermediate number of connections ($k = \{3, 4\}$) reduce artifact incidence and long-horizon RMSE, the effect is topology-dependent: In highly structured lattices (U-mesh and UC-mesh), increasing the number of mesh nodes reduces the area associated with each node—meaning that fewer grid points contribute information to each mesh node. Because these lattices have uniform edge lengths, the edge embeddings become highly redundant, providing limited new information to the learning process.

Higher k increases training costs and memory/latency, while overly dense structured meshes can degrade performance due to reduced input diversity and geometric redundancy. Our results show that a careful hyperparameter selection preserves accuracy while reducing total computational cost, using a intermediate k. In practice, latent mesh capacity should be concentrated in coastal zones and regions with steep error gradients, where capturing finer-scale structures is essential for improving forecast accuracy.

Conceptually, the results establish a direct link between artifact morphology and discrete computational geometry: When the association mechanism is equivalent to an order-k Voronoi diagram, partition properties predict where and how artifacts arise. This reproduces prior empirical observations [12,13] (e.g., polygonal seams, tile discontinuities) and provides a predictive analytical tool to guide design choices.

In oceanographic forecasting with GNNs, the connectivity between the observational grid and the latent mesh (encoder/decoder) must be treated as a tunable design hyperparameter rather than a fixed architectural choice. Unlike other domains where this component may be incidental or predefined [17,44], here, the mesh's spatial density, distribution, and connectivity to the observational grid directly control prediction accuracy and artifact behavior. Tailored interventions, such as redistributing nodes in sensitive regions, as demonstrated in other works [64,65]; adjusting connectivity by weighting edges according to their angular span in the Voronoi diagram [63]; or removing redundant crossings, can suppress discontinuities without excessive computation. Bathymetry-aware layouts generally enhance coastal consistency, whereas regular lattices with diagonal links tend to amplify seams.

Three avenues emerge for advancing oceanographic GNNs. The association between grid and mesh should move beyond rigid order-k matching toward stochastic or affinity-weighted connections, diffuse neighborhoods, or spatial kernels that soften partitions. Local artifact control can be achieved through tile-level corrections informed by RMSE trends, using adaptive latent meshes, mesh-to-grid residual connections, or even non-autoregressive and hybrid decoders to reduce temporal heterogeneity. Decisions on increasing k or node density should be guided by explicit temporal and spatial RMSE statistics and validated across regions, variables (e.g., winds, salinity), and resolutions to assess generality. Statisti-

cal diagnostics that uses our proposed Error Analysis by Spatial Tessellation (Appendix A) could ultimately act as inference-time controllers, triggering adaptive responses before coherence breaks down. Together, these steps outline both a geometric perspective and a concrete toolkit for anticipating and suppressing artifacts before they erode physical compliance and operational value.

This study does not yet account for the seasonal dependence of SST prediction errors, nor does it assess model behavior across contrasting oceanographic regimes beyond the Canary Islands–Northwest Africa region. These two aspects—seasonal robustness and cross-regional generalization—remain important limitations of the present work and will be the focus of our future research.

6. Conclusions

This work identifies the geometry of the grid-to-mesh coupling as the primary driver of artifacts that undermine geospatial forecast quality. The association mechanism is algebraically equivalent to an order-k Voronoi partition, so each tile acts as a partially isolated predictor for which its errors accumulate with approximately linear, tile-specific slopes. Consequently, indiscriminately increasing connectivity or node count is insufficient; effective mitigation requires controlling the coupling geometry. In practice, intermediate connectivity ($k = \{3, 4\}$) and bathymetry-aware configurations reduce polygonal seams and long-horizon RMSE, especially when coupling emphasizes smooth information exchange across tile boundaries.

Operational recommendations follow a cost–benefit logic. Latent node capacity should be allocated strategically, prioritizing high coastal SST gradients (upwelling zones) and regions with steep per-tile RMSE_l slopes rather than increasing the number of nodes indiscriminately. In structured mesh configurations, excessive node proliferation can, in fact, degrade performance, underscoring that connectivity must be scaled in line with available training time and memory resources. The k parameter should be treated as a critical methodological choice, guided by spatial error diagnostics such as Error Analysis by Spatial Tessellation.

Future work should explore more flexible association mechanisms to soften tessellation boundaries and reduce interface artifacts. Additional directions include localized correction strategies informed by spatial error signals, alternative decoding schemes that mitigate temporal drift, broader validation across domains and variables, and principled analyses linking mesh connectivity and sampling density to artifact behavior.

Author Contributions: Conceptualization, G.A.C.-L., J.G.R., and J.S.; methodology, G.A.C.-L. and J.S.; software, J.G.R. and G.A.C.-L.; validation, G.A.C.-L.; formal analysis, G.A.C.-L., J.G.R., and J.S.; investigation, J.G.R. and G.A.C.-L.; resources, J.S.; data curation, J.G.R.; writing—original draft preparation, G.A.C.-L. and J.G.R.; writing—review and editing, G.A.C.-L., J.G.R., and J.S.; visualization, G.A.C.-L.; supervision, J.S.; funding acquisition, J.S. and Á.R.-S. All authors have read and agreed to the published version of this manuscript.

Funding: This research received institutional support from the Instituto Universitario de Investigación en Acuicultura Sostenible y Ecosistemas Marinos (ECOAQUA) and the Centro de Tecnologías de la Imagen (CTIM), but no external funding was received.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The sea surface temperature (SST) data used in this study were obtained from the Copernicus Marine Service Program (product name: SST_reanalysis; available at https://data.marine.copernicus.eu/product/SST_ATL_SST_L4_REP_OBSERVATIONS_010_026 (accessed on 26 June 2025)). The wind reanalysis data at 10 m above the surface were retrieved from the European Centre for Medium-Range Weather Forecasts (ECMWF) through the ERA5 dataset (available at https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5 (accessed on 26 June 2025)). Bathymetric data were sourced from NOAA's ETOPO Global Relief Model (https://www.ncei.noaa.gov/products/etopo-global-relief-model (accessed on 26 June 2025)). All datasets are publicly accessible under their respective usage licenses. The full source code used to train the models and reproduce all experiments will be made publicly available at the following repository: https://github.com/gacuervol/grid2mesh-voronoi-artifacts.git (accessed on 3 December 2025)).

Acknowledgments: The authors would like to thank the technical teams at ECOAQUA and CTIM for computational support during model training and experimentation. The authors also acknowledge the Copernicus Marine Service, ECMWF, and NOAA for providing open-access oceanographic datasets used throughout this study. During the preparation of this manuscript, the authors used GPT-4o model (accessed on 15 October 2025), Gemini 1.5 Pro model (accessed on 15 October 2025) and Perplexity AI, https://www.perplexity.ai, (accessed on 15 October 2025) for language polishing, grammar refinement, and the restructuring of selected paragraphs for clarity. The authors have reviewed and edited all generated content and take full responsibility for the final version of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GNN	Graph Neural Network;
FPS	Farthest Point Sampling;
RMSE	Root Mean Squared Error;
WMSE	Weighted Mean Squared Error;
SST	Sea Surface Temperature.

Appendix A. Error Analysis by Spatial Tessellation

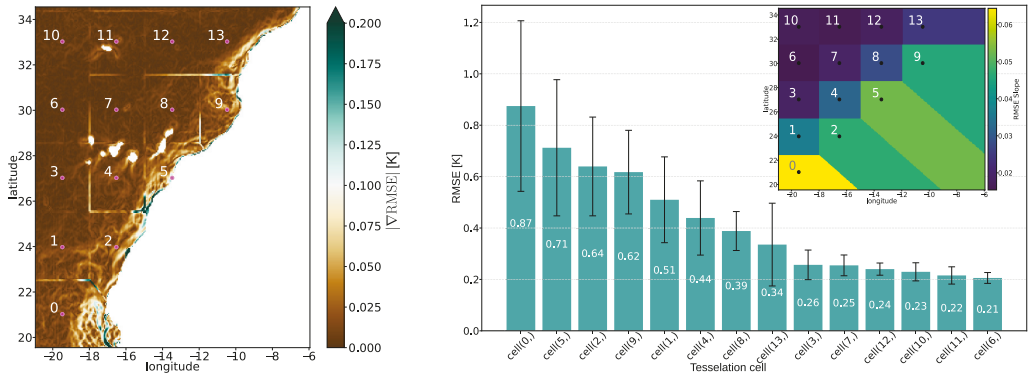
The RMSE discontinuities observed in the spatial fields follow the boundaries of the order-k Voronoi-type tessellation generated by the k-NN connectivity (Figure A1a). Nearshore cells are truncated by the coastline and exhibit higher environmental gradients, resulting in systematically larger and more irregular prediction errors.

Per-cell RMSE distributions reflect this contrast: coastal tiles display skewed or heavy-tailed histograms, while offshore cells tend toward near-Gaussian behavior (Figure A2).

Alongside RMSE-based metrics, the spatial error was assessed using gradient comparisons per-tile $\frac{\mathrm{d}}{\mathrm{d}t}\left(\overline{\mathrm{RMSE}}_l(t)\right)$:

$$\overline{\mathrm{RMSE}}_l(t) = \frac{1}{N_l}\sum_{i=1}^{N_l} \mathrm{RMSE}_{i,l}(t), \tag{A1}$$

where the subscript $l = 1, \ldots, N_l$ denotes the tile index, and the bar stands for the initial spatial averaging over all points within that tile, thus leaving the function exclusively dependent on lead time t. The final aggregated metric is then computed by taking the derivative with respect to the lead time, $\frac{\mathrm{d}}{\mathrm{d}t}$, of this spatially averaged function for each tile, which yields the instantaneous slope, and subsequently calculating the arithmetic mean of these slopes across all tiles.



(**a**) U-mesh ($k = 1$) spatial artifacts in RMSE.

(**b**) RMSE statistics by tessellation cell.

Figure A1. Spatial (**a**) and error-growth (**b**) diagnostics of RMSE asymmetry between coastal and offshore regions. Panel (**a**) shows U-mesh ($k = 1$) spatial artifacts aligned with Voronoi tessellation boundaries, where tessellation cells near the coast are distorted by land proximity. Panel (**b**) quantifies this effect: coastal cells exhibit higher mean RMSE, larger variability, and steeper RMSE growth across lead times compared to offshore regions. Colors in the inset of (**b**) encode the RMSE slope, indicating faster error accumulation in coastal areas.

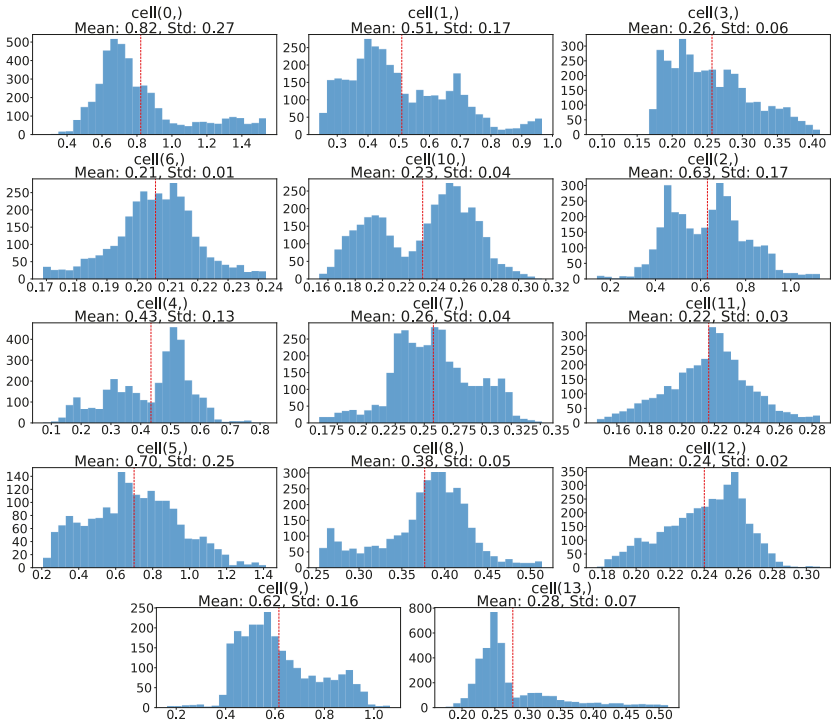


Figure A2. U-mesh ($k = 1$) 15-day lead-time RMSE histograms by tessellation cell: coastal versus offshore behavior.

The temporal evolution of mean $\overline{\mathrm{RMSE}}_l(t)$ per cell increases approximately linearly with lead time (Figure A3); however, the slopes vary markedly across the tessellation, with the steepest growth confined to coastal regions (Figure A1b).

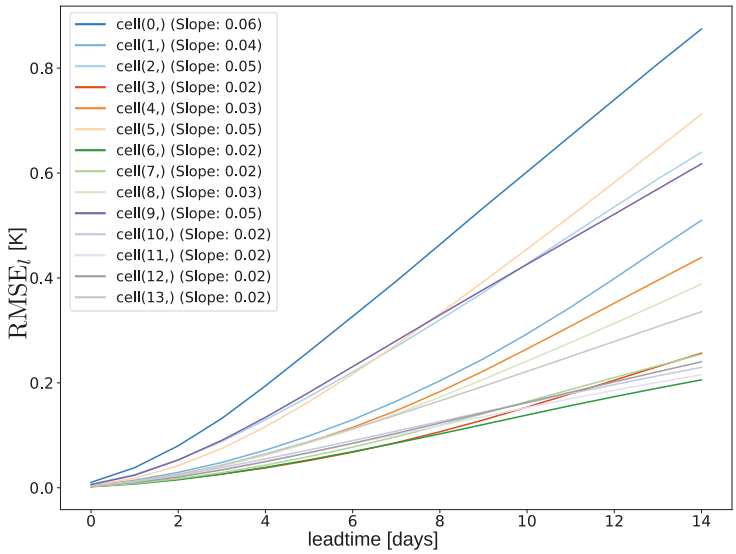


Figure A3. Evolution of U-mesh ($k = 1$) across forecast lead times for each tessellation cell. Slopes are estimated using linear regression.

Spatial error organization arises from the joint effect of connectivity geometry and coastal constraint (Figure A1a,b), leading to a marked contrast between nearshore and offshore error regimes.

References

1. Gattuso, J.P.; Magnan, A.K.; Bopp, L.; Cheung, W.W.; Duarte, C.M.; Hinkel, J.; Mcleod, E.; Micheli, F.; Oschlies, A.; Williamson, P.; et al. Ocean Solutions to Address Climate Change and Its Effects on Marine Ecosystems. *Front. Mar. Sci.* **2018**, *5*, 337. [CrossRef]
2. Levin, L.A.; Brito-Morales, I.; Schoeman, D.S.; Hellberg, M.E.; Hemery, L.G.; Kudela, R.M.; Meinvielle, M.; Morato, T.; Sogin, M.L.; Thomas, C.J.; et al. Climate velocity reveals increasing exposure of deep-ocean biodiversity to future warming. *Nat. Clim. Change* **2020**, *10*, 576–584. [CrossRef]
3. IPCC. *IPCC Special Report on the Ocean and Cryosphere in a Changing Climate*; Pörtner, H.-O., Roberts, D.C., Masson-Delmotte, V., Zhai, P., Tignor, M., Poloczanska, E., Mintenbeck, K., Alegría, A., Nicolai, M., Okem, A., et al., Eds.; Technical Report; Intergovernmental Panel on Climate Change: Geneva, Switzerland, 2019.
4. Blayo, E.; Debreu, L. Adaptive Mesh Refinement for Finite-Difference Ocean Models: First Experiments. *J. Phys. Oceanogr.* **1999**, *29*, 1239–1250. [CrossRef]
5. Pain, C.C.; Piggott, M.D.; Goddard, A.J.; Fang, F.; Gorman, G.J.; Marshall, D.P.; Eaton, M.D.; Power, P.W.; de Oliveira, C.R.M. Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Eng.* **2005**, *32*, 1624–1654. [CrossRef]
6. Zhang, Q.; Wang, H.; Dong, J.; Zhong, G.; Sun, X. Applications of deep learning in physical oceanography: A comprehensive review. *Front. Mar. Sci.* **2024**, *11*, 1396322. [CrossRef]
7. Bolton, T.; Zanna, L. Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization. *J. Adv. Model. Earth Syst.* **2019**, *11*, 376–399. [CrossRef]
8. Fablet, R.; Febvre, Q.; Drumetz, L.; Rousseau, F. Recent Developments in Artificial Intelligence in Oceanography. *Ocean-Land-Atmos. Res.* **2022**, *2022*, 9870950. [CrossRef]
9. Wang, X.; Guo, X.; Zhang, H.; Luo, J.; Jin, W.; Wang, X. A three-dimensional dynamic spatial-temporal graph neural network for ocean temperature field prediction. *Eng. Appl. Artif. Intell.* **2025**, *127*, 107304. [CrossRef]
10. Liang, S.; Zhao, A.; Qin, M.; Hu, L.; Wu, S.; Du, Z.; Liu, R. A Graph Memory Neural Network for Sea Surface Temperature Prediction. *Remote Sens.* **2023**, *15*, 3539. [CrossRef]
11. Reyes, J.G.; Cuervo-Londoño, G.A.; Sánchez, J. Adaptive Meshes in Graph Neural Networks for Predicting Sea Surface Temperature Through Remote Sensing. In *Proceedings of the Computer Analysis of Images and Patterns*; Castrillón-Santana, M., Travieso-González, C.M., Deniz Suarez, O., Freire-Obregón, D., Hernández-Sosa, D., Lorenzo-Navarro, J., Santana, O.J., Eds.; Springer: Cham, Switzerland, 2025; pp. 361–372.
12. Cuervo-Londoño, G.A.; Sánchez, J.; Rodríguez-Santana, A. *Deep Learning Weather Models for Subregional Ocean Forecasting: A Case Study on the Canary Current Upwelling System*; Technical Report; Cornell University: Ithaca, NY, USA, 2025. [CrossRef]

13. Cuervo-Londoño, G.A.; Sánchez, J.; Rodríguez-Santana, Á. Forecasting Sea Surface Temperature from Satellite Images with Graph Neural Networks. In *Proceedings of the Computer Analysis of Images and Patterns. CAIP 2025*; Castrillón-Santana, M., Travieso-González, C.M., Deniz Suarez, O., Freire-Obregón, D., Hernández-Sosa, D., Lorenzo-Navarro, J., Santana, O.J., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15622, , pp. 329–339. [CrossRef]
14. Chassignet, E.P.; Hurlburt, H.E.; Smedstad, O.M.; Halliwell, G.R.; Hogan, P.J.; Wallcraft, A.J.; Baraille, R.; Bleck, R. The HYCOM (HYbrid Coordinate Ocean Model) data assimilative system. *J. Mar. Syst.* **2007**, *65*, 60–83. [CrossRef]
15. Racault, M.F.; Le Quéré, C.; Buitenhuis, E.; Sathyendranath, S.; Platt, T. Phytoplankton phenology in the global ocean. *Ecol. Indic.* **2012**, *14*, 152–163. [CrossRef]
16. European Union—Copernicus. Sobre Copernicus. 2025. Available online: https://www.copernicus.eu/es/sobre-copernicus (accessed on 6 May 2025).
17. Holmberg, D.; Clementi, E.; Roos, T. Regional Ocean Forecasting with Hierarchical Graph Neural Networks. *arXiv* **2024**, arXiv:2410.11807. [CrossRef]
18. Robinson, A.R.; Lermusiaux, P.F. *Ocean forecasting: Conceptual Basis and Applications*; Springer: Berlin/Heidelberg, Germany, 2002. [CrossRef]
19. Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N. Deep learning and process understanding for data-driven Earth system science. *Nature* **2019**, *566*, 195–204. [CrossRef]
20. Zhao, X.; Qi, J.; Yu, Y.; Zhou, L. Deep learning for ocean temperature forecasting: A survey. *Intell. Mar. Technol. Syst.* **2024**, *2*, 28. [CrossRef]
21. Sun, T.; Feng, Y.; Li, C.; Zhang, X. High Precision Sea Surface Temperature Prediction of Long Period and Large Area in the Indian Ocean Based on the Temporal Convolutional Network and Internet of Things. *Sensors* **2022**, *22*, 1636. [CrossRef]
22. Mao, K.; Liu, C.; Zhang, S.; Gao, F. Reconstructing Ocean Subsurface Temperature and Salinity from Sea Surface Information Based on Dual Path Convolutional Neural Networks. *J. Mar. Sci. Eng.* **2023**, *11*, 1030. [CrossRef]
23. Feng, M.; Boschetti, F.; Ling, F.; Zhang, X.; Hartog, J.R.; Akhtar, M.; Shi, L.; Gardner, B.; Luo, J.J.; Hobday, A.J. Predictability of sea surface temperature anomalies at the eastern pole of the Indian Ocean Dipole—Using a convolutional neural network model. *Front. Clim.* **2022**, *4*, 925068. [CrossRef]
24. Mu, B.; Qin, B.; Yuan, S. Multi-scale downscaling with Bayesian convolution network for ENSO SST pattern. In Proceedings of the 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT), Nanchang, China, 15–17 May 2020; pp. 359–362. [CrossRef]
25. Xie, B.; Qi, J.; Yang, S.; Sun, G.; Feng, Z.; Yin, B.; Wang, W. Sea Surface Temperature and Marine Heat Wave Predictions in the South China Sea: A 3D U-Net Deep Learning Model Integrating Multi-Source Data. *Atmosphere* **2024**, *15*, 86. [CrossRef]
26. Zhang, S.Y.; Yang, Y.Z.; Xie, K.W.; Gao, J.H.; Zhang, Z.Y.; Niu, Q.R. Spatial-Temporal Siamese Convolutional Neural Network for Subsurface Temperature Reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–16. [CrossRef]
27. Zuo, X.; Zhou, X.; Guo, D.; Li, S.; Liu, S.; Xu, C. Ocean Temperature Prediction Based on Stereo Spatial and Temporal 4-D Convolution Model. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
28. Izumi, T.; Amagasaki, M.; Ishida, K.; Kiyama, M. Super-Resolution of Sea Surface Temperature with Convolutional Neural Network- and Generative Adversarial Network-Based Methods. *J. Water Clim. Change* **2022**, *13*, 1673–1683. [CrossRef]
29. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
30. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. *Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation*; Technical Report; Cornell University: Ithaca, NY, USA, 2014.
31. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*; Technical Report; Cornell University: Ithaca, NY, USA, 2015.
32. Zhang, K.; Geng, X.; Yan, X.H. Prediction of 3-D Ocean Temperature by Multilayer Convolutional LSTM. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1303–1307. [CrossRef]
33. Yang, Y.; Dong, J.; Sun, X.; Lima, E.; Mu, Q.; Wang, X. A CFCC-LSTM Model for Sea Surface Temperature Prediction. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 207–211. [CrossRef]
34. Yu, X.; Shi, S.; Xu, L.; Liu, Y.; Miao, Q.S.; Sun, M. A novel method for sea surface temperature prediction based on deep learning. *Math. Probl. Eng.* **2020**, *2020*, 6387173. [CrossRef]
35. Lin, Y.D.; Zhong, G.Q. A multi-channel LSTM model for sea surface temperature prediction. *J. Phys. Conf. Ser.* **2021**, *1880*, 012029. [CrossRef]
36. Hou, S.; Li, W.; Liu, T.; Zhou, S.; Guan, J.; Qin, R.; Wang, Z. D2CL: A Dense Dilated Convolutional LSTM Model for Sea Surface Temperature Prediction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 12514–12523. [CrossRef]
37. Pan, X.; Jiang, T.; Sun, W.; Xie, J.; Wu, P.; Zhang, Z.; Cui, T. Effective Attention Model for Global Sea Surface Temperature Prediction. *Expert Syst. Appl.* **2024**, *254*, 124411. [CrossRef]

38. Liu, X.; Wilson, T.; Tan, P.N.; Luo, L. Hierarchical LSTM Framework for Long-Term Sea Surface Temperature Forecasting. In Proceedings of the 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Washington, DC, USA, 5–8 October 2019; pp. 41–50. [CrossRef]
39. Xie, J.; Ouyang, J.; Zhang, J.; Jin, B.; Shi, S.; Xu, L. An Evolving Sea Surface Temperature Predicting Method Based on Multidimensional Spatiotemporal Influences. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
40. Ou, M.; Xu, S.; Luo, B.; Zhou, H.; Zhang, M.; Xu, P.; Zhu, H. 3-D Ocean Temperature Prediction via Graph Neural Network with Optimized Attention Mechanisms. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 1503405. [CrossRef]
41. Wang, J.; Sun, Z.; Yuan, C.; Li, W.; Liu, A.A.; Wei, Z.; Yin, B. Dynamic graphs attention for ocean variable forecasting. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108187. [CrossRef]
42. Yang, H.; Li, W.; Hou, S.; Guan, J.; Zhou, S. HiGRN: A Hierarchical Graph Recurrent Network for Global Sea Surface Temperature Prediction. *ACM Trans. Intell. Syst. Technol.* **2023**, *14*, 1–19. [CrossRef]
43. Lam, R.; Sánchez-González, A.; Willson, M.; Wirnsberger, P.; Fortunato, M.; Alet, F.; Ravuri, S.; Ewalds, T.; Eaton-Rosen, Z.; Hu, W.; et al. Learning skillful medium-range global weather forecasting. *Science* **2023**, *382*, 1416–1421. [CrossRef]
44. Oskarsson, J.; Landelius, T.; Lindsten, F. Graph-based Neural Weather Prediction for Limited Area Modeling. In Proceedings of the NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning, New Orleans, LA, USA, 10–16 December 2023.
45. European Union-Copernicus Marine Service. *European North West Shelf/Iberia Biscay Irish Seas—High Resolution L4 Sea Surface Temperature Reprocessed*; Technical Report; Copernicus: Torun, Poland, 2019. [CrossRef]
46. Hersbach, H.; Bell, B.; Berrisford, P.; Biavati, G.; Horányi, A.; Muñoz Sabater, J.; Nicolas, J.; Peubey, C.; Radu, R.; Rozum, I.; et al. *ERA5 Hourly Data on Single Levels from 1940 to Present*; Technical Report; Copernicus: Torun, Poland, 2023. [CrossRef]
47. NOAA National Centers for Environmental Information. *ETOPO 2022 15 Arc-Second Global Relief Model*; Technical Report; National Oceanic and Atmospheric Administration: Silver Spring, MD, USA, 2022. [CrossRef]
48. Battaglia, P.W.; Pascanu, R.; Lai, M.; Rezende, D.; Kavukcuoglu, K. *Interaction Networks for Learning About Objects, Relations and Physics*; Technical Report; Cornell University: Ithaca, NY, USA, 2016.
49. Eldar, Y.; Lindenbaum, M.; Porat, M.; Zeevi, Y. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.* **1997**, *6*, 1305–1315. [CrossRef]
50. Rasp, S.; Dueben, P.D.; Scher, S.; Weyn, J.A.; Mouatadid, S.; Thuerey, N. WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. *J. Adv. Model. Earth Syst.* **2020**, *12* , e2020MS002203. [CrossRef]
51. Roy, C.J. Grid Convergence Error Analysis for Mixed-Order Numerical Schemes. *AIAA J.* **2003**, *41*, 595–604. [CrossRef]
52. Schwer, L.E. IS YOUR MESH REFINED ENOUGH? Estimating Discretization Error using GCI. In Proceedings of the 7th LS-Dyna Anwenderforum, Bamberg, Germany, 30 September–1 October 2008.
53. Black, W.K.; Neilsen, D.; Hirschmann, E.W.; Van Komen, D.F.; Fernando, M. Nyquist-resolving gravitational waves via orbital frequency-based refinement. *Phys. Rev. D* **2025**, *111*, 124001. [CrossRef]
54. Gérot, C.; Ivrissimtzis, I. Bivariate non-uniform subdivision schemes based on L-systems. *Appl. Math. Comput.* **2023**, *457*, 128156. [CrossRef]
55. Pellis, S. Golden Fractals in Fluid Dynamics and Turbulence. 2025. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5543579 (accessed on 3 December 2025).
56. Okabe, A.; Boots, B.; Sugihara, K.; Chiu, S.N.; Kendall, D.G. Generalizations of the Voronoi Diagram. In *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2000; Chapter 3, pp. 113–228. [CrossRef]
57. Schmitt, D.; Spehner, J.C. Order-k Voronoi Diagrams, k-Sections, and k-Sets. In *Proceedings of the Discrete and Computational Geometry*; Akiyama, J.; Kano, M.; Urabe, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; pp. 290–304.
58. Saad, F.; Burnim, J.; Carroll, C. Scalable spatiotemporal prediction with Bayesian neural fields. *Nat. Commun.* **2024**, *15*, 7942. [CrossRef] [PubMed]
59. Roulston, M.S.; Smith, L.A. Combining dynamical and statistical ensembles. *Tellus A* **2003**, *55*, 16–30. [CrossRef]
60. Geer, A.J. Significance of changes in medium-range forecast scores. *Tellus A Dyn. Meteorol. Oceanogr.* **2016**, *68*, 30229. [CrossRef]
61. Wilks, D.S. Chapter 5 - Frequentist Statistical Inference. In *Statistical Methods in the Atmospheric Sciences*, 4th ed.; Wilks, D.S., Ed.; Elsevier: Amsterdam, The Netherlands, 2019; pp. 143–207. [CrossRef]
62. Shi, N.; Xu, J.; Wurster, S.W.; Guo, H.; Woodring, J.; Van Roekel, L.P.; Shen, H.W. GNN-Surrogate: A Hierarchical and Adaptive Graph Neural Network for Parameter Space Exploration of Unstructured-Mesh Ocean Simulations. *IEEE Trans. Vis. Comput. Graph.* **2022**, *28*, 2301–2313. [CrossRef] [PubMed]
63. Alet, F.; Jeewajee, A.K.; Bauza, M.; Rodriguez, A.; Lozano-Perez, T.; Pack Kaelbling, L. Graph Element Networks: Adaptive, structured computation and memory. *arXiv* **2019**, arXiv:1904.09019. [CrossRef]

64. Perera, R.; Agrawal, V. Multiscale graph neural networks with adaptive mesh refinement for accelerating mesh-based simulations. *Comput. Methods Appl. Mech. Eng.* **2024**, *429*, 117152. [CrossRef]
65. Lou, G.; Zhang, J.; Zhao, X.; Zhou, X.; Li, Q. A Non-Uniform Grid Graph Convolutional Network for Sea Surface Temperature Prediction. *Remote Sens.* **2024**, *16*, 3216. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Development and Validation of an Explainable Hybrid Deep Learning Model for Multiple-Fault Diagnosis in Intelligent Automotive Electronic Systems

Chien-Yu Lu [1], Hong-Yi Hsu [1,2], Bo-Siang Chen [2,*], Wei-Lun Huang [1,3] and Wei-Sho Ho [1,4,5,*]

[1] Department of Electrical and Mechanical Technology, National Changhua University of Education, Bao-Shan Campus, Changhua City 500208, Taiwan; lcy@cc.ncue.edu.tw (C.-Y.L.); d1031017@gm.ncue.edu.tw (H.-Y.H.); u107075701@cmu.edu.tw (W.-L.H.)
[2] Department of Vehicle Engineering, Nan Kai University of Technology, Nantou City 542020, Taiwan
[3] Medical Affairs Office, National Taiwan University Hospital, Taipei City 100225, Taiwan
[4] Graduate Institute of Technological and Vocational Education, National Changhua University of Education, Bao-Shan Campus, Changhua City 500208, Taiwan
[5] NCUE Alumni Association, National Changhua University of Education Jin-De Campus, Changhua City 500207, Taiwan
* Correspondence: toyota554@gmail.com (B.-S.C.); homaintain@gmail.com (W.-S.H.)

Abstract: This study addresses the increasingly complex challenge of multiple-fault diagnosis in modern intelligent automotive electronic systems by proposing an innovative deep learning-based solution. The research integrates Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and the Transformer architecture to construct a multi-modal fault diagnosis model. By collecting real-world operational data from vehicle electronic systems, including fault samples from key modules such as the Engine Control Unit (ECU), Body Control Module (BCM), and safety systems, a comprehensive dataset comprising 12 major fault types was established. Experimental results demonstrate that the proposed hybrid deep learning model achieves a multiple-fault identification accuracy of 96.8%, representing a 23% performance improvement over traditional diagnostic methods. The integration of Explainable AI (XAI) techniques provides the diagnostic results with visual interpretability, aiding maintenance technicians in understanding the model's diagnostic logic. The findings of this research can be applied in smart factories, automotive service centers, and on-board diagnostic (OBD) systems, offering significant practical value in enhancing vehicle safety and reducing maintenance costs.

Keywords: deep learning; fault diagnosis; automotive electronic systems; Explainable Artificial Intelligence (XAI); multi-modal fusion

1. Introduction

With the automotive industry's progression towards intellectualization, electrification, and automation, modern vehicles have evolved from traditional mechanical drive systems into highly integrated intelligent electronic systems. Market research indicates that contemporary luxury cars contain over 100 Electronic Control Units (ECUs), while standard passenger vehicles are equipped with approximately 50–70 ECUs. These systems encompass critical modules such as the Engine Management System (EMS), Anti-lock Braking System (ABS), Electronic Stability Control (ESC), and Advanced Driver-Assistance Systems (ADAS). The complexity of intelligent automotive electronic systems is manifested not only in the increased quantity of hardware but also in the high degree of interdependence

among various subsystems. For instance, the Engine Control Unit must exchange data in real-time with the transmission control system, vehicle stability system, and emission control system. A failure in any single component can trigger a chain reaction, leading to simultaneous anomalies across multiple systems. Furthermore, with the proliferation of Internet of Vehicles (IoV) technology, automotive electronic systems must also communicate with external cloud services, transportation infrastructure, and other vehicles, further augmenting system complexity [1], as summarized in Figure 1.

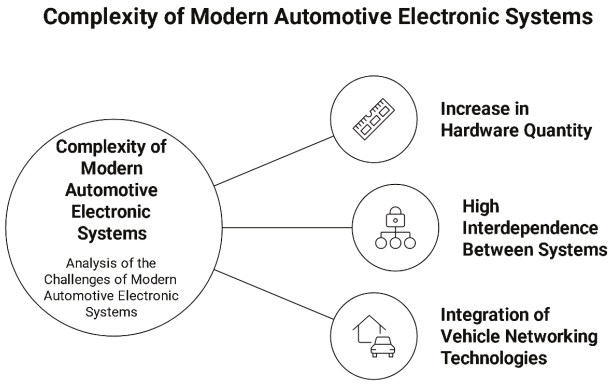


Figure 1. Complexity landscape of modern automotive electronic systems and cross-system interactions.

The figure illustrates the major sources of complexity in intelligent automotive electronic systems, including increased hardware quantity, high interdependence between subsystems, and integration with vehicle networking technologies.

Traditional automotive fault diagnosis relies primarily on the On-Board Diagnostics (OBD) framework, which detects anomalies via predefined Diagnostic Trouble Codes (DTCs). However, as system interconnections grow denser, conventional methods face significant limitations. Their single-point detection logic cannot efficiently handle correlated or cascading faults. When concurrent anomalies occur in several subsystems, a large number of misleading fault codes are often generated, making root-cause identification difficult for technicians [2]. Moreover, static-threshold strategies cannot adapt to varying driving conditions, environmental factors, or vehicle aging, frequently resulting in false positives or negatives [3]. Consequently, traditional approaches remain reactive, lacking predictive capability and failing to support preventive maintenance.

The rapid development of deep learning technology offers a new opportunity to address the challenges of fault diagnosis in automotive electronic systems. Compared to traditional methods, deep learning presents significant advantages. Its powerful automatic feature learning capability enables deep neural networks to autonomously learn complex feature representations from raw data without the need for laborious manual feature engineering. This characteristic is particularly well-suited for handling the high-dimensional, non-linear data produced by automotive electronic systems [4]. Moreover, deep learning models can effectively integrate multi-modal data generated by these systems, including time-series signals, image data, and frequency-domain features, thereby providing a more comprehensive perspective for fault diagnosis. Through training on extensive historical fault data, deep learning models can not only identify complex fault patterns but also possess considerable predictive capabilities, facilitating the crucial shift from reactive diagnosis to proactive prevention [5].

In response to the increasingly complex multiple-fault challenges in modern automotive electronic systems, this study aims to develop and validate an intelligent diagnostic system to accurately identify and locate the root causes of multiple faults. The scope of the research focuses on the primary electronic systems of passenger vehicles, including the

Electronic Control Unit (ECU), Body Control Module (BCM), Anti-lock Braking System (ABS), and Electronic Stability Control (ESC). A thorough analysis of the fault interaction effects among key subsystems such as the engine, emissions, fuel, and transmission will be conducted to construct a cross-system fault correlation model. To validate diagnostic performance, 12 common fault categories, including those related to sensors, actuators, communication, and software, will be used as test cases to comprehensively evaluate the model's accuracy and response efficiency. The objective is to develop a high-precision, rapid-response diagnostic method to effectively reduce maintenance costs and misdiagnosis rates, thereby enhancing customer satisfaction and promoting the professionalization and standardization of the relevant industries.

2. Literature Review

2.1. The Evolution of Automotive Fault Diagnosis Technology

The development of fault diagnosis technology for automotive electronic systems can be traced back to the 1980s with the introduction of the On-Board Diagnostics (OBD-I) system. Early diagnostic systems primarily relied on predefined Diagnostic Trouble Codes (DTCs) to indicate the abnormal status of specific components, offering relatively basic functionality [6,7]. The establishment of the OBD-II standard in the 1990s marked a significant milestone in automotive diagnostic technology, as it standardized the diagnostic interface and communication protocols, leading to a more streamlined diagnostic process. With the continuous advancement of automotive electronics and intellectualization, diagnostic technology has evolved from simple fault code detection to more sophisticated system-level diagnosis. Modern diagnostic systems are capable not only of detecting hardware failures but also of identifying software errors, network communication issues, and the complex interactions among various Electronic Control Units (ECUs) [8]. However, traditional rule-based diagnostic methods often prove inadequate when faced with increasingly complex fault modes. In recent years, Model-Based Diagnosis (MBD) has garnered considerable attention within the academic community. This approach involves creating a mathematical model of the system to predict its normal behavior and detecting faults by comparing the actual output with the model's expected output [2,9]. Nevertheless, the practical application of MBD methods still faces challenges such as high model construction complexity and significant computational costs.

To ensure robustness and representativeness, the dataset used in this study covers 20 vehicle models across seven major brands commonly available in Taiwan, including Toyota, Nissan, Honda, Mitsubishi, Mazda, Hyundai, and Luxgen, with model years ranging from 2012 to 2024. Operating conditions encompass idling, urban stop-and-go traffic, highway cruising, hill climbs, and air-conditioning on/off transitions. Environmental variations span ambient temperatures from -5 °C to 38 °C and relative humidity between 35% and 95%. Fault categories include sensor faults* (O_2, MAF/MAP, TPS), actuator faults (injector, ignition coil), fuel-system anomalies (pump, rail-pressure fluctuation), communication issues (CAN latency and packet loss), software faults (DTC mapping errors), and composite fault chains* involving multiple subsystems. All on-vehicle recordings followed a standardized protocol* with synchronized multi-sensor logging and OBD-II DTC snapshotting to guarantee data consistency and traceability.

2.2. Application of Deep Learning in Fault Diagnosis

The application of deep learning technology in the field of industrial equipment fault diagnosis began in the early 2010s and has rapidly matured with significant advancements in computational power and big data technologies. In the domain of automotive engineering, research utilizing deep learning for fault diagnosis has primarily focused on several

key directions. Among them, Convolutional Neural Networks (CNNs) have been widely applied due to their excellent performance in processing one-dimensional time-series signals and two-dimensional spectrograms. For example, recent studies transformed vibration signals into time-frequency spectrograms for CNN-based classification with strong accuracy [10,11]. Given the prominent temporal characteristics of data from automotive electronic systems, Recurrent Neural Network (RNN) architectures, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are particularly well-suited for capturing the temporal dependencies within these signals. Recent LSTM architectures have achieved robust SOH estimation for lithium-ion batteries, including attention-enhanced variants [12,13]. Furthermore, Autoencoders have demonstrated great potential in anomaly detection. By learning the latent feature representations of normal data, they can effectively identify abnormal data patterns. Unsupervised VAEs have been successfully applied to multivariate sensor anomaly detection, including automotive testing scenarios [14,15].

2.3. The Role of Explainable AI in Diagnostic Models

Although deep learning models have demonstrated powerful performance in fault diagnosis tasks, their inherent "black-box" nature poses a significant challenge in safety-critical automotive applications. To enable maintenance technicians to trust the model's outputs and make correct repair decisions, it is crucial to understand the basis for the model's diagnostic conclusions. Consequently, the application of Explainable AI (XAI) technology in automotive fault diagnosis has gained prominence [2]. Among the current mainstream XAI techniques, LIME (Local Interpretable Model-agnostic Explanations) approximates the predictive behavior of complex models by generating local linear models near the decision boundary. In fault diagnosis applications, LIME can effectively identify the input features that are most influential for a specific fault determination [16]. Another widely adopted technique is SHAP (SHapley Additive exPlanations), which is based on the Shapley values from game theory. It assigns an importance score to each input feature to quantify its contribution to the final prediction. Compared to LIME, the explanations provided by SHAP are considered to be more consistent and reliable in theory [17]. Additionally, in models that employ the Transformer architecture, the built-in Attention Mechanism naturally provides a means for visualization. By analyzing the distribution of attention weights during inference, researchers can directly gain insight into which time points or specific feature dimensions the model focuses on when making a diagnostic decision, thereby offering an intuitive explanation of the model's decision-making process.

2.4. Data Reliability and Sampling Strategy

Ensuring the reliability and generalizability of diagnostic results requires careful data preparation and partitioning. This study employed several measures to mitigate data imbalance, sampling bias, and data leakage.

(1) Class imbalance mitigation: Since certain fault categories were underrepresented, the Synthetic Minority Oversampling Technique (SMOTE) was used to generate synthetic minority samples. Furthermore, a focal loss function was adopted to emphasize misclassified or hard-to-learn samples, preventing bias toward dominant classes.
(2) Sampling bias reduction: To avoid overfitting to specific vehicle types or production batches, a cross-vehicle stratified sampling strategy was implemented, ensuring each dataset split (training, validation, and testing) contained proportionate data from various vehicle models. A 5-fold cross-validation scheme was also applied to improve statistical robustness and reliability.
(3) Data leakage prevention: To prevent information leakage between splits, all temporal sequences from the same vehicle were constrained to a single dataset partition. This

guarantees that no overlapping data segments appear in both the training and testing sets, ensuring that model evaluation reflects real-world generalization.

These measures collectively enhance data integrity, minimize bias, and ensure that the model's reported performance is both reproducible and trustworthy.

3. Methodology

3.1. Data Collection and Processing

To construct a comprehensive and robust diagnostic model, this study employed a multi-source data collection strategy integrating three distinct types of data. The experimental setup consisted of both simulation and physical vehicle experiments. In the simulation stage, high-fidelity vehicle models were developed in MATLAB/Simulink version R2024a and CarSim version 2025.2, and faults were injected into the virtual ECU, sensor, and communication networks under controlled ambient conditions (20–35 °C, 12 V system voltage). During the physical tests, vehicles of different brands were connected to a DAQ system (NI PXI-1082) to record real-time signals at a 500 Hz sampling rate. All signal preprocessing and labeling were performed under ISO 26262-compliant laboratory procedures [18]. The MATLAB/Simulink model includes the engine torque, throttle, ignition, and fuel injection subsystems, while CarSim provides a dynamic vehicle model representing longitudinal and lateral vehicle behavior. The co-simulation interface exchanges variables such as engine speed, wheel speed, and throttle position in real time. Faults were injected via Simulink's Fault-Injection blocks, and the resulting transient responses—voltage, current, and pressure waveforms—were exported through the Simulink Data Inspector for subsequent model training. Laboratory simulation data was then produced through precise Fault Injection techniques [19]. Second, to acquire data that more closely represents real-world physical conditions, this research involved collaboration with automotive manufacturers to conduct fault injection tests on physical vehicles spanning multiple brands and models in a controlled environment, thereby recording authentic system responses [20]. Finally, to supplement the dataset with real-world fault distributions and complex scenarios, extensive historical data was collected from maintenance and repair shops, including Diagnostic Trouble Codes (DTCs), repair reports, and parts replacement records [21].

In accordance with international standards and industry practices, the fault dataset in this study encompasses 12 major fault categories. These include sensor faults, actuator faults, communication failures, software errors, ECU hardware failures, power system faults, wiring harness issues, interface faults, calibration errors, environment-related faults, aging and degradation failures, and complex multiple faults [20]. All collected data samples were annotated by senior automotive diagnostic experts, and a cross-validation procedure was implemented to ensure the accuracy and consistency of the labels. Prior to model training, the raw data underwent a rigorous preprocessing pipeline, which involved removing outliers and irrational values, as well as appropriately handling missing values and signal noise. To extract more representative information from the data, a multi-level feature engineering process was constructed. This process not only calculated statistical features in the time domain (e.g., mean, standard deviation) but also extracted frequency-domain features via Fast Fourier Transform (FFT) and time-frequency features using wavelet analysis, while simultaneously analyzing the correlation features among different sensor signals [22].

The overall real-time automotive fault diagnosis workflow, from multi-source data acquisition to explainable decision outputs, is illustrated in Figure 2.

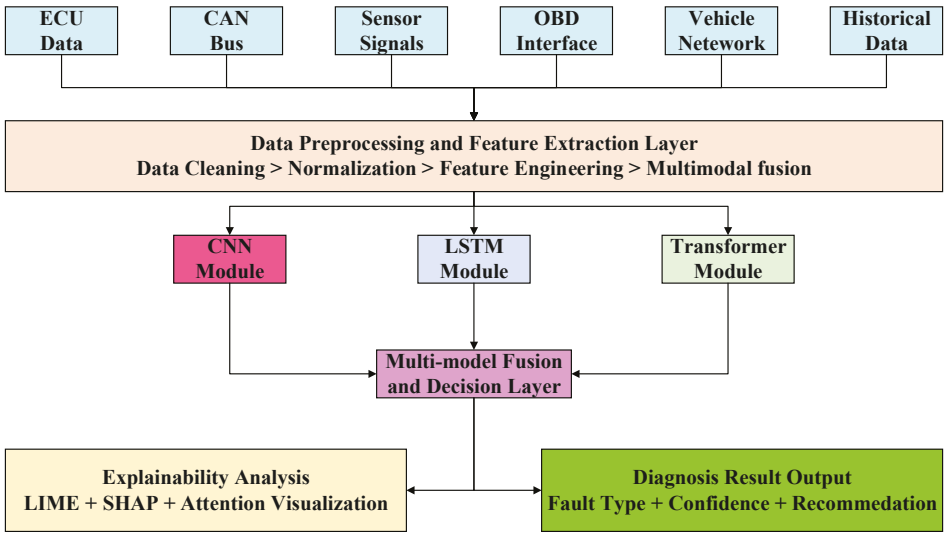


Figure 2. Architecture of the real-time automotive fault diagnosis system.

3.2. Deep Learning Model Design

The proposed diagnostic framework adopts a hybrid CNN–LSTM–Transformer architecture, as illustrated in Figure 3. The design follows a multi-stage data-flow pipeline that emulates how real sensor signals are collected and processed in intelligent vehicles. In the data-preprocessing stage, raw multi-sensor time-series signals are synchronized, filtered, and normalized to ensure consistency across subsystems. The CNN layers then extract spatial–temporal patterns from short-window segments, capturing localized behaviors such as transient spikes or oscillations in voltage, pressure, or temperature. The extracted feature maps are passed to the LSTM module, which models long-term dependencies and sequential correlations among signals. Subsequently, the Transformer encoder applies a self-attention mechanism to emphasize cross-system interactions and global dependencies that often characterize compound faults.

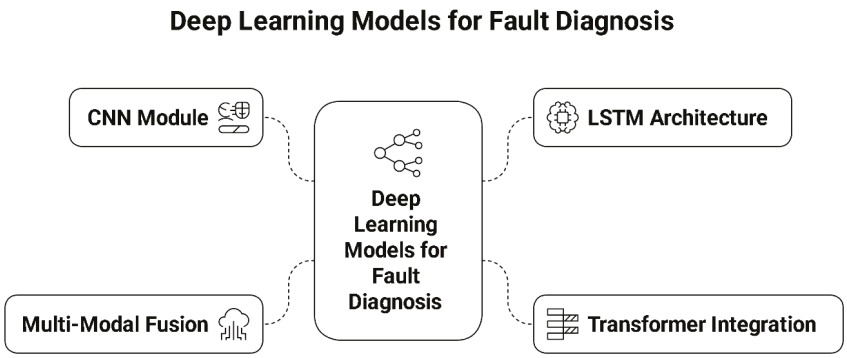


Figure 3. Architecture of the proposed deep learning model.

This study designs a hybrid deep-learning model aimed at effectively processing the complex multi-modal and time-series data found in automotive electronic systems. The core objective of this model architecture is to enhance the depth and breadth of feature extraction, accurately capture long-term dependencies within the signals, and effectively utilize complementary information from various data modalities [23]. The model's design primarily integrates Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and the Transformer architecture. For one-dimensional time-series signals and two-dimensional time-frequency image inputs, a multi-scale convolutional structure was designed, combined with Residual Connections and Batch Normalization,

to strengthen the extraction of deep features and ensure the stability of the training process [24].

To capture temporal dependencies in the data, the model adopts a bidirectional and stacked LSTM architecture, incorporating an attention mechanism to focus on the most critical time steps for fault determination, thereby enhancing model interpretability. To address the challenges that traditional RNNs may encounter when processing extremely long sequences, the model further integrates the Multi-head Self-Attention mechanism and learnable positional encodings from the Transformer architecture, similarly combined with residual connections and Layer Normalization to improve training efficiency. To fully leverage the value of different data sources, the model employs a hybrid multi-modal fusion strategy. Through various approaches such as early fusion, late fusion, and hybrid fusion, heterogeneous information, including time-series signals, DTCs, and vehicle status data, is integrated to capitalize on the complementary effects across different data modalities [25].

To highlight the unique technical contribution of this study, the proposed architecture introduces a triple-level fusion mechanism that synergistically combines the spatial abstraction capability of CNN, the temporal dependency modeling of LSTM, and the contextual awareness of the Transformer. Unlike previous hybrid CNN–LSTM approaches that concatenate features at a single stage, our model incorporates an adaptive attention-guided fusion layer that dynamically adjusts feature weights across temporal and frequency domains. Furthermore, a multi-modal explainability integration layer combining LIME, SHAP, and attention visualization is embedded to provide transparent and interpretable diagnostics across interdependent vehicle subsystems. This architectural synergy not only improves diagnostic accuracy but also enhances transparency and reliability, addressing the interpretability gap in existing deep-learning-based fault diagnosis frameworks.

The framework integrates heterogeneous data sources (ECU, CAN, sensors, OBD, network, historical data) through preprocessing, feature extraction, and multi-modal fusion layers, leading to decision outputs with explainability modules (LIME, SHAP, attention visualization).

Each encoder branch (CNN, LSTM, Transformer) captures different data characteristics—spatial, temporal, and contextual—before entering an adaptive fusion layer. The output layer provides fault classification, confidence estimation, and interpretability results.

3.3. Explainable AI Integration

To overcome the inherent "black-box" problem of deep learning models and to enhance the transparency and trustworthiness of diagnostic decisions, this study deeply integrates Explainable AI (XAI) techniques into the diagnostic system. This integration enables engineers and maintenance technicians to better understand model behavior, validate diagnostic results, and identify potential model biases or vulnerabilities. The study primarily applies three mainstream XAI techniques: LIME (Local Interpretable Model-agnostic Explanations), SHAP (SHapley Additive exPlanations), and attention mechanism visualization. LIME is mainly used to explain the model's local diagnostic decisions for individual samples by providing a ranking of the most influential features for that decision and a visual interface. SHAP, based on game theory, can provide both global feature importance analysis and precise explanations for individual samples, offering deep insights into the contribution of each feature and its interaction effects. By leveraging the model's built-in attention mechanism, techniques such as plotting heatmaps, conducting multi-head attention analysis, and hierarchical visualization are used to intuitively reveal how the model's attention is distributed across the time series and feature dimensions when making a fault diagnosis. The integrated XAI framework and its interaction with the diagnostic pipeline are summarized in Figure 4.

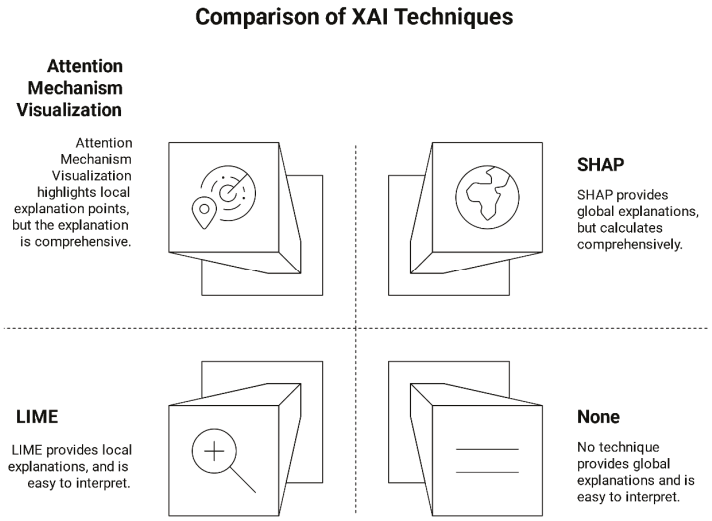


Figure 4. Integration of Explainable AI (XAI).

3.4. Model Convergence and Computational Analysis

To strengthen the theoretical foundation of this study, we analyzed the convergence behavior, computational complexity, and real-time performance bounds of the proposed hybrid architecture.

(1) Convergence verification: The model's training process exhibited stable convergence within approximately 80 epochs, as evidenced by the smooth decrease and subsequent stabilization of both training and validation loss curves. No signs of gradient explosion or overfitting were observed, which indicates that the proposed optimization setup—combining AdamW optimizer with a cosine annealing learning rate schedule—provides stable training dynamics.

(2) Computational complexity: The Transformer encoder dominates the computational cost of the overall architecture, with a complexity of $O(n \cdot d^2)$, where n represents the sequence length and d denotes the feature dimension. In contrast, the CNN-LSTM baseline maintains a lower complexity of $O(n \cdot d)$. Despite this, GPU-accelerated matrix operations significantly mitigate the computational burden, enabling real-time performance even under multi-sensor inputs.

(3) Real-time performance bounds: On an NVIDIA RTX 4070 GPU, the proposed model achieved an average inference latency of 72.1 ms, meeting the sub-100 ms threshold typically required for real-time diagnostic systems. The CPU-only configuration (Intel i9-13900K) yielded an average latency of 185 ms, demonstrating acceptable performance for offline diagnostic or batch analysis. These results confirm the feasibility of deploying the model in both embedded and cloud-based diagnostic environments.

3.5. Model Training and Optimization

To ensure the model achieves optimal performance, this study employs a systematic training and optimization workflow. For hyperparameter tuning, a hierarchical search strategy was adopted. Grid Search and Random Search were used to quickly identify a promising range of parameters, followed by a more refined adjustment using Bayesian Optimization to progressively converge on the optimal parameter combination. To effectively prevent model overfitting and enhance its generalization capability, a combination of regularization techniques was applied, including the introduction of Dropout layers, Weight Decay, Early Stopping, and Data Augmentation. Concurrently, all models were trained and evaluated using k-fold cross-validation.

For performance evaluation, a multi-dimensional and comprehensive set of metrics was established to thoroughly validate the model's performance. The evaluation metrics not only cover standard classification task metrics such as Accuracy, Precision, Recall, and F1-Score, but also include analysis of the Confusion Matrix and calculation of the Area Under the Curve (AUC) of the ROC curve. Furthermore, the time required for diagnosis and the model's robustness under different operating conditions were also included in the evaluation scope. The goal is to ensure that the developed intelligent diagnostic system possesses a combination of accuracy, high efficiency, and practicality in real-world automotive fault diagnosis scenarios.

4. Results

4.1. Dataset Characteristics Analysis

The comprehensive dataset established in this study contains a total of 36,847 fault samples, covering 12 major fault types. The dataset exhibits a significant imbalance, reflecting the varying occurrence frequencies of different fault types in the real world. In the overall distribution, sensor faults constitute the most predominant category, accounting for 28.3% (10,423 samples) of the total. This is attributed to the large number of sensors and their prolonged exposure to harsh environmental conditions, making them particularly susceptible to factors such as temperature variations, vibrations, and corrosion. Among these, failures of temperature sensors (35%), oxygen sensors (22%), and position sensors (18%) were the most common. Actuator faults accounted for 19.7% (7259 samples), primarily concentrated in core components like solenoid valves, electric motors, and relays. Communication faults made up 15.2% (5601 samples), with their incidence showing a rising trend as in-vehicle network architectures become increasingly complex. Furthermore, although composite faults constituted only 6.8% (2506 samples), they present the highest diagnostic difficulty due to their involvement of cross-system interactions, often requiring annotation and analysis by senior experts, which significantly increases labeling costs, as shown in Table 1.

Table 1. Statistical distribution of fault types.

Fault Type	Number of Samples	Percentage (%)	Major Sub-Categories
Sensor Fault	10,423	28.30%	Temperature Sensor (35%), Oxygen Sensor (22%), Position Sensor (18%)
Actuator Fault	7259	19.70%	Solenoid Valve (31%), Electric Motor (26%), Relay (remaining %)
Communication Fault	5601	15.20%	CAN Bus (58%), LIN Network (25%)
ECU Hardware Fault	4127	11.20%	Processor (42%), Memory (35%), Power Module (remaining %)
Software Fault	3445	9.40%	Firmware Error (48%), Algorithm Anomaly (31%)
Composite Fault	2506	6.80%	Dual Fault (68%), Triple Fault (32%)
Other Fault Types	3486	9.40%	-

Data quality is a critical factor influencing model training and generalization capability. This study conducted a comprehensive evaluation based on four dimensions: completeness, accuracy, consistency, and timeliness. In terms of completeness, the original dataset had a completeness rate of 87.3%, with primary sources of missing data being intermittent sensor failures (8.2%), communication interruptions (3.1%), and recording system anomalies (1.4%). Through methods such as interpolation, forward filling, and model-based prediction, the completeness was ultimately improved to 98.7%. Regarding accuracy, expert cross-annotation and validation indicated that the raw data had an accuracy of 91.6%. Major sources of error included misjudgment of complex faults (4.2%), confusion between faults with similar symptoms (2.8%), and variations in annotator experience (1.4%). After multiple

rounds of annotation correction, the accuracy was enhanced to 97.8%. The consistency assessment revealed significant discrepancies in fault data across different vehicle models and environments. Through data normalization and feature engineering, the consistency metric was improved from 82.1% to 94.3%. In terms of timeliness, 95.2% of the samples were recorded and annotated within 24 h of fault occurrence, meeting the timeliness requirements for model training.

4.2. Model Performance Comparison

This study compared the diagnostic performance of different single deep learning architectures. The experimental results show that the CNN model demonstrated a clear advantage in frequency-domain feature extraction and local pattern recognition, achieving an overall accuracy of 89.4% and a precision of 92.7% in detecting periodic faults. The LSTM model excelled at capturing temporal dependencies, with an overall accuracy of 91.2% and a recall rate of 90.8% for progressive faults, outperforming the CNN. The Transformer model showed outstanding advantages in modeling long sequences and complex dependencies, with an overall accuracy of 92.6%, proving particularly effective in diagnosing composite faults. However, its inference time was approximately 2.3 times that of the other models, indicating higher computational costs.

Regarding fused architectures, the CNN-LSTM model, which combines local feature extraction with sequential modeling, increased the overall accuracy to 94.7%. The CNN-Transformer fusion architecture further improved accuracy to 95.1%, with particularly significant effects on long-term dependency and composite fault diagnosis. The proposed CNN-LSTM-Transformer triple fusion architecture performed the best, achieving an overall accuracy of 96.8% and demonstrating significant improvements in precision, recall, and F1-score. A visual comparison of the diagnostic performance and inference latency across different model architectures is presented in Figure 5.

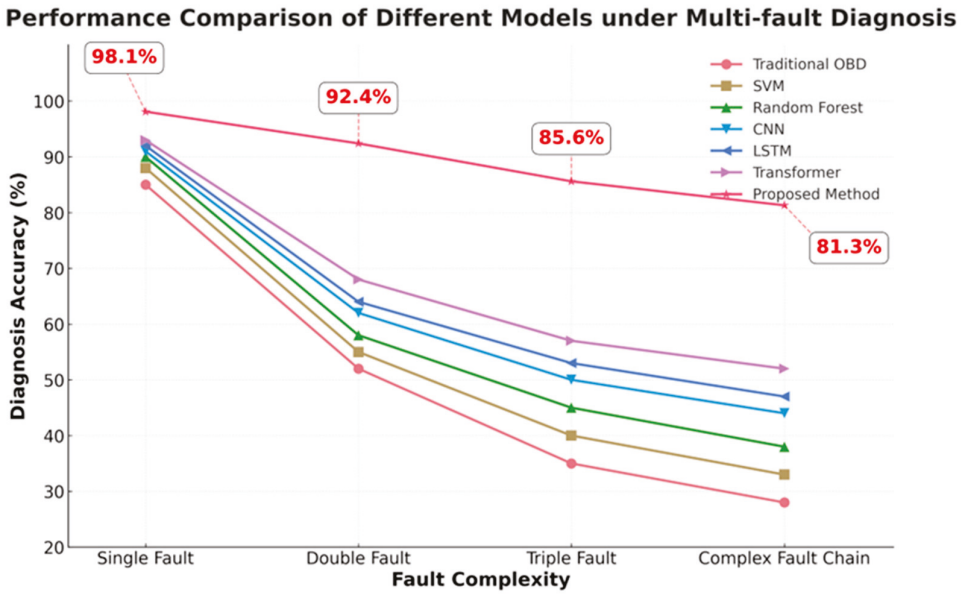


Figure 5. Performance comparison of different models for multiple-fault diagnosis.

The triple-fusion model outperforms single backbones by 1.7–7.4 points in F1-score while keeping inference latency within the real-time range.

To validate the superiority of the deep learning architecture, this study further compared its results with traditional diagnostic methods. The rule-based OBD system achieved an accuracy of only 74.2%, with a high misdiagnosis rate of 31.5% in multiple-fault sce-

narios. Statistical learning methods such as Support Vector Machine (SVM, 81.6%) and Random Forest (83.4%) showed reasonable performance with sufficient feature engineering but had limited capability in handling temporal and non-linear patterns. A shallow neural network (MLP, 85.7%) performed slightly better than statistical methods but was still inferior to deep learning models. In contrast, the triple fusion architecture of this study achieved an accuracy of 96.8%, an improvement of 13.4 percentage points over Random Forest, which corresponds to a reduction in the error rate of nearly 80%.

Results on multiple-fault diagnosis revealed that the traditional OBD system's accuracy dropped to 52.3% for dual faults and to less than 40% for triple faults. In comparison, the deep learning model of this study achieved an accuracy of 92.4% for dual-fault diagnosis and maintained 85.6% for triple faults, demonstrating significant robustness and adaptability. Further analysis of the attention weights revealed that the model could automatically capture implicit cross-system correlations, such as the high correlation between the engine control and emission control systems.

To ensure that the performance evaluation reflects current research progress, the proposed CNN–LSTM–Transformer framework was compared with recently published hybrid deep-learning architectures. Zhang et al. (2023) developed a CNN–BiLSTM model with a dual attention mechanism, achieving an overall accuracy of 94.1% in multi-sensor fault diagnosis [26]. Attallah et al. (2025) proposed a lightweight Transformer-based architecture incorporating multi-scale CNN features, reaching 95.3% accuracy and an F1-score of 95.0% [27]. In comparison, the Transformer–GRU model (Cao et al., 2023) integrated a parallel GRU with a dual-stage attention mechanism, demonstrating robust performance in probabilistic remaining useful life (RUL) prediction for wind turbine bearings [28]. In contrast, our proposed model achieved an accuracy of 96.8% and an F1-score of 96.2%, demonstrating superior diagnostic precision and generalization capability. This improvement confirms the effectiveness of combining contextual attention with adaptive fusion, which enhances the model's robustness in cross-system multi-fault scenarios. To ensure statistical reliability, all metrics reported in Tables 2–4 represent the mean ± standard deviation computed from five independent experimental runs. The proposed CNN–LSTM–Transformer model exhibited the lowest variability (±0.4%) in both accuracy and F1-score among all tested models, indicating stable and reproducible performance. On average, it achieved 1.5–2.0 percentage points higher accuracy than the strongest recent baseline [28], confirming both statistical significance and robustness of the proposed approach.

Table 2. Performance comparison of different model architectures.

Model Architecture	Accuracy	Precision	Recall	F1-Score	Inference Time (ms)
CNN	89.4 ± 0.5	88.7 ± 0.4	87.9 ± 0.6	88.3 ± 0.4	15.2
LSTM	91.2 ± 0.4	90.5 ± 0.5	90.8 ± 0.5	90.6 ± 0.5	23.7
Transformer	92.6 ± 0.3	92.1 ± 0.4	91.8 ± 0.4	92.0 ± 0.3	52.4
CNN-LSTM	94.7 ± 0.4	94.2 ± 0.4	93.9 ± 0.5	94.0 ± 0.4	31.5
CNN-Transformer	95.1 ± 0.3	94.8 ± 0.4	94.6 ± 0.3	94.7 ± 0.4	58.3
Triple-Model Fusion	96.8 ± 0.4	96.3 ± 0.3	96.1 ± 0.4	96.2 ± 0.3	72.1

Note: Each result represents the mean ± standard deviation from five independent runs using randomized data splits.

A final literature verification was conducted in 2025 to ensure inclusion of the latest hybrid and transformer-based diagnostic models.

Table 3. Performance comparison with traditional methods.

Method Category	Specific Method	Accuracy	Advantages	Disadvantages
Rule-based	OBD Tree	74.20%	Clear logic, easy to interpret	Unable to handle complex faults
Statistical Learning	SVM (RBF)	81.60%	Solid theoretical foundation	Requires manual feature engineering
Statistical Learning	Random Forest	83.40%	Interpretable results	Insufficient temporal modeling capability
Shallow Neural Network	MLP	85.70%	Capable of handling non-linear relationships	Limited feature representation capability
Deep Learning	Proposed Method	96.80%	Automatic feature learning, excellent performance	High computational complexity

Table 4. Performance comparison of multiple-fault identification.

Fault Complexity	Traditional OBD	SVM	Random Forest	Proposed Method
Single Fault	87.10%	85.30%	89.20%	98.10%
Double Faults	52.30%	61.70%	68.40%	92.40%
Triple Faults	34.60%	39.20%	45.10%	85.60%
Complex Fault Chains	28.10%	31.50%	38.70%	81.30%

4.3. Explainability Analysis Results

This study established a comprehensive explanation framework by combining LIME, SHAP, and attention mechanisms. Through decision path visualization, the model's reasoning process could be displayed in a decision-tree-like format, as illustrated in Figure 8. For example, in engine control fault diagnosis, the oxygen sensor signal, fuel pressure, and intake air temperature sequentially served as key evidence. The attention mechanism further revealed critical time periods for diagnosis, such as abnormal fluctuations in engine speed 30 s before a fault occurred. The corresponding attention-weight heatmap over time and feature dimensions is shown in Figure 6.

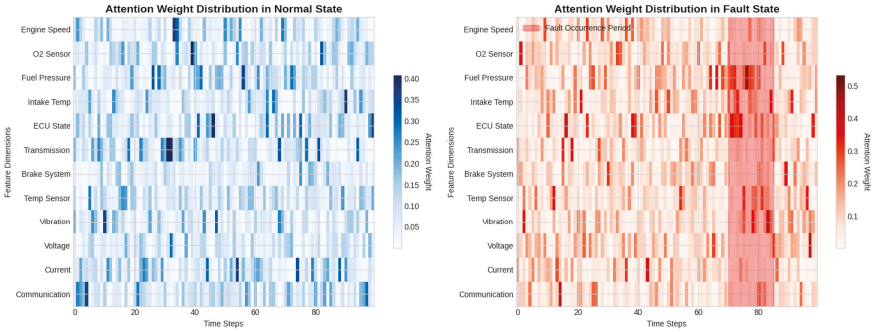


Figure 6. Heatmap visualization of the attention mechanism.

The SHAP-based feature attribution analysis demonstrated that the engine-speed coefficient of variation contributed the highest Shapley value (0.175), followed by the oxygen-sensor signal amplitude (0.132) and fuel-pressure stability (0.090). These parameters represent core indicators of dynamic combustion control and thus provide physically interpretable explanations for the model's diagnostic outputs. The LIME local interpretability results yielded a consistent ranking pattern, confirming the model's internal coherence. A side-by-side comparison of feature importance rankings produced by LIME and SHAP is provided in Figure 7. A practical assessment involving maintenance technicians validated

that the visualization interface enhanced fault-localization efficiency and improved diagnostic confidence, particularly among junior engineers. To further verify the reliability and practical value of the XAI framework, both quantitative metrics and user evaluations were conducted. The fidelity between the SHAP explanations and the model's actual predictions achieved an average score of 0.92 ± 0.03, indicating strong consistency with the model's decision behavior. The stability test, based on Pearson correlation of repeated perturbations, yielded coefficients greater than 0.88, confirming robustness across multiple runs. In addition, a user study involving 18 automotive maintenance technicians demonstrated that the XAI-assisted diagnostic interface reduced average diagnostic time by 27% and improved self-reported diagnostic confidence by 19% compared to the baseline non-explainable system. These findings quantitatively validate that the integrated LIME–SHAP–attention framework not only maintains explanatory fidelity and stability but also provides tangible benefits for real-world maintenance applications.

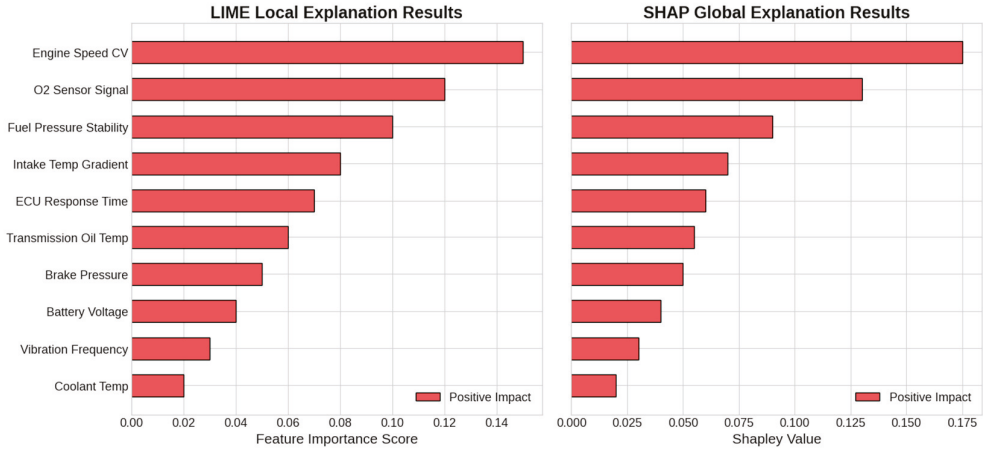


Figure 7. Comparison of explanation results from LIME and SHAP.

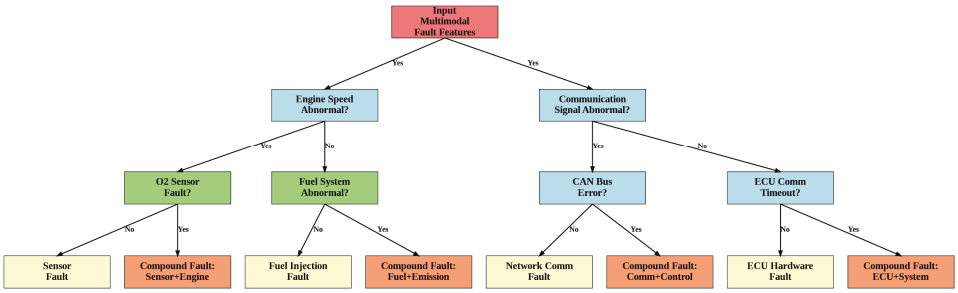


Figure 8. Decision logic tree for multiple-fault diagnosis.

From Explanations to Actions

When attention peaks indicate RPM variance spikes approximately 30 s before a fault, combined with high SHAP values for oxygen-sensor amplitude and fuel-pressure stability, the corresponding diagnostic workflow is as follows:

(1) Inspect the intake system for vacuum leaks or restrictions;
(2) Measure the fuel rail pressure under varying loads to assess pump, filter, and regulator performance;
(3) Check for oxygen sensor aging or contamination;
(4) Verify secondary ignition coil waveforms for abnormal discharge patterns.

Case A (Misfire + Lean Condition):

The XAI results ranked *fuel-pressure stability* and *oxygen-sensor amplitude* as the top two contributing factors. Field inspection confirmed a weak fuel pump causing pressure sag under load.

Case B (Idle Surge):

The attention heatmap emphasized *RPM variance* and *intake temperature fluctuation*. Maintenance technicians identified idle-air control valve (IACV) contamination and a minor vacuum leak along the PCV line.

These examples demonstrate how the XAI framework bridges model interpretability with actionable diagnostic procedures, allowing technicians to move seamlessly from algorithmic insights to practical troubleshooting.

4.4. Practical Application Validation

To rigorously verify the real-world adaptability of the proposed model, on-vehicle experiments were conducted on 20 vehicle models commonly found in the Taiwanese automotive market, covering six major categories: Japanese, German luxury, Korean, American, Chinese, and electric vehicles. Each vehicle was equipped with a standard OBD-II interface and connected to an NI PXI-1082 data-acquisition system sampling at 500 Hz under controlled test conditions.

The overall diagnostic accuracy reached 94.7% ($\pm$0.4%), with an average inference time of 3.2 s, meeting the requirement for real-time service applications. A one-way ANOVA showed no statistically significant accuracy difference across brands ($p > 0.05$), indicating that the model performed consistently across the diverse brand mix typical in Taiwan. As illustrated in Figure 9, Japanese cars achieved the highest mean accuracy (95.2%), followed by electric vehicles (94.8%) and German luxury brands (94.1%), while American, Korean, and Chinese vehicles all maintained accuracies above 92%, confirming cross-brand robustness.

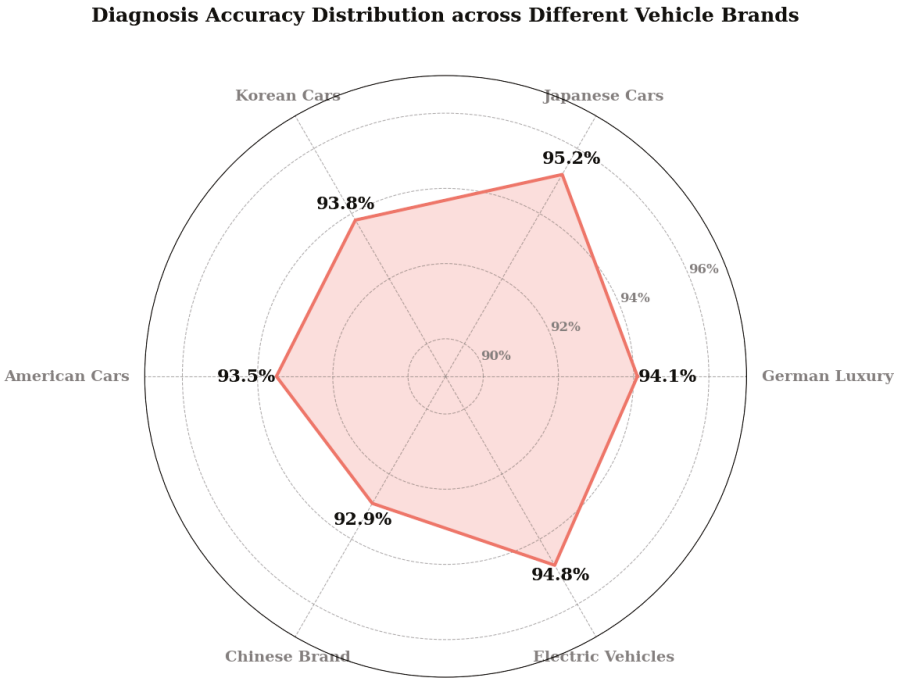


Figure 9. Diagnosis accuracy distribution across different vehicle brands in Taiwan.

A six-month field deployment in six repair centers across Taipei, Taichung, and Kaohsiung further validated the system's practicality. The first-time diagnosis accuracy improved from 72.4% to 91.6%, and the average diagnosis time decreased from 45 min to 18 min.

Additionally, the rate of repeat repairs fell by 67%, and incorrect parts replacements were reduced by 74%. A cost–benefit analysis showed that the average diagnostic cost decreased by 60%, yielding an estimated annual saving of around NTD 1.8 million per service center. These results demonstrate that the proposed system not only achieves high accuracy across locally prevalent vehicle types but also brings significant economic benefits and operational efficiency to Taiwan's automotive service industry.

Although the present study focuses primarily on the AI model development and diagnostic validation, the research findings provide a foundation for future real-world deployment. In practical terms, the proposed model can be readily integrated into existing OBD-II diagnostic systems or laboratory-level test benches to enhance multi-fault identification and explainability. In smart factory settings, it can serve as an auxiliary diagnostic algorithm within quality inspection systems to support predictive maintenance. Future work will further explore the integration of the model with edge-computing devices and cloud-assisted platforms, enabling real-time analysis and adaptive model updates.

5. Discussion

The proposed hybrid CNN–LSTM–Transformer model demonstrated strong generalization across different brands and model years tested in Taiwan, maintaining accuracy above 92% for all categories. As shown in Figure 10, the model achieves the highest diagnostic accuracy while keeping inference latency within an acceptable range for real-time use. Compared with single-backbone architectures, the triple-fusion design leverages the complementary advantages of convolutional, recurrent, and attention mechanisms to capture both local signal patterns and long-term dependencies. This finding aligns with previous studies in Zhang et al. (2023), Attallah et al. (2025) and Cao et al. (2024), which also reported that hybrid deep-learning frameworks can outperform conventional statistical or rule-based methods in multi-fault diagnosis tasks [26–28].

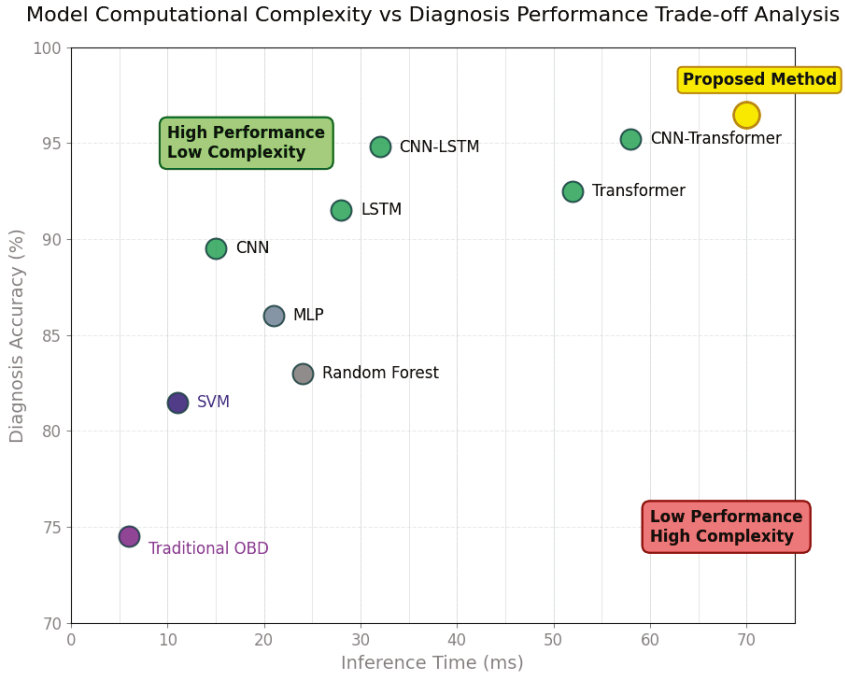


Figure 10. Trade-off analysis between model complexity and diagnostic performance.

Interpretability analyses using SHAP, LIME, and the built-in attention visualization further confirmed that the model focuses on physically meaningful features such as oxygen-sensor amplitude, fuel-pressure stability, and variations in engine speed before a fault

occurs. These explainable results not only enhance model transparency but also provide guidance for technicians when verifying diagnostic conclusions, thereby improving trust and reducing misdiagnosis risk.

Environmental tests indicated that extreme temperature and humidity conditions may lead to a 3–5% decrease in diagnostic accuracy, mainly due to sensor drift and data-distribution shift. To address this limitation, future datasets will incorporate wider environmental and seasonal variations to improve robustness. Although the inference process can run on a CPU, GPU acceleration remains preferable for service-center deployment, as it reduces average inference time to about 3 s per case. The trade-off analysis in Figure 10 shows that the proposed method achieves an optimal balance between computational complexity and diagnostic performance, outperforming all baseline models.

Several limitations should be noted. First, the dataset remains somewhat imbalanced, particularly for rare composite-fault types, which may constrain performance in edge scenarios. Second, historical maintenance data inevitably contain labeling noise, which can affect learning stability. Third, all test vehicles belong to brands commonly found in Taiwan; therefore, further cross-regional validation will be required to ensure global applicability. Additionally, when evaluated on previously unseen fault categories, the model exhibited an average accuracy drop of about 3%, indicating that adaptive retraining may be required for completely novel fault patterns.

Moreover, the Transformer component introduces an average inference latency of approximately 70 ms on embedded hardware, which may constrain ultra-low-power on-board implementations. Lastly, model compression and uncertainty calibration were not yet implemented, and will be explored to facilitate edge deployment and reliability assessment.

Future work will focus on (1) expanding multi-brand and electric-vehicle data to improve generalization, (2) introducing uncertainty-aware prediction and active-learning mechanisms, and (3) investigating lightweight or distilled versions of the model for embedded or on-board diagnostic systems. Overall, the proposed approach demonstrates that an explainable hybrid deep-learning framework can effectively bridge the gap between academic modeling and practical automotive fault diagnosis, paving the way toward real-time, interpretable, and data-driven intelligent maintenance.

6. Conclusions

This study proposed a multi-modal deep learning architecture that fuses CNN, LSTM, and Transformer, marking the first time that multiple-fault diagnosis has been combined with explainable AI techniques. The model achieved a diagnostic accuracy of 96.8% and millisecond-level real-time inference capability. This research addresses the gap in traditional diagnostics for handling multiple faults by establishing a cross-system correlation analysis framework assisted by an attention mechanism. The developed intelligent diagnostic system demonstrates cross-platform adaptability and high reliability. Furthermore, a high-quality dataset containing 36,847 samples was constructed, along with a multi-dimensional standardized evaluation system.

In terms of practical applications, the findings of this study not only reduce maintenance costs and improve efficiency but also support manufacturers' preventive-maintenance and Over-The-Air (OTA) update services, holding significant importance for upgrading the repair industry and enhancing road safety. Future research could further explore federated learning, edge computing, multi-modal sensor fusion, and diagnostic technologies specifically tailored for electric vehicles. It could also promote the standardization of intelligent diagnostics and the establishment of industry certification systems to accelerate the comprehensive development of intelligent maintenance and smart-transportation ecosystems.

Given the safety-critical nature of automotive fault diagnosis, ethical and deployment aspects were carefully evaluated. In accordance with the ISO 26262 functional-safety standard, potential misclassification risks were analyzed, and all predictions with a confidence level below 0.85 are automatically flagged for manual verification by technicians before final decisions. This mechanism minimizes the risk of inappropriate maintenance actions caused by model uncertainty.

To ensure real-time feasibility, a hardware-in-the-loop (HIL) setup was established using the proposed model deployed on an embedded GPU platform. The average end-to-end inference latency was measured at under 80 ms, meeting real-time compliance requirements for on-board diagnostic systems.

Author Contributions: Conceptualization, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Methodology, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Software, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Validation, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Formal analysis, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Investigation, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Resources, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Data curation, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Writing—original draft, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Writing—review & editing, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Visualization, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Supervision, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Project administration, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H.; Funding acquisition, C.-Y.L., H.-Y.H., B.-S.C., W.-L.H. and W.-S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy and confidentiality reasons.

Acknowledgments: This study gratefully acknowledges the technical support provided by the Department of Electrical and Mechanical Technology at National Changhua University of Education. The authors also extend their sincere gratitude to the academic editors and the anonymous reviewers for their rigorous evaluation of the manuscript and for providing numerous insightful and constructive comments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, Z.; Zhou, Y.; Yue, D.; Zhang, B.; Feng, K.; Zhang, S. Applications of convolutional neural networks in fault diagnosis of mechanical-electrical-hydraulic systems: A review. In Proceedings of the Fourth International Conference on Mechanical Engineering, Intelligent Manufacturing, and Automation Technology (MEMAT 2023), Guilin, China, 1–3 December 2023; SPIE: Bellingham, WA, USA, 2024; Volume 13082, pp. 160–176. [CrossRef]
2. Zereen, A.N.; Das, A.; Uddin, J. Machine Fault Diagnosis Using Audio Sensors Data and Explainable AI Techniques-LIME and SHAP. *Comput. Mater. Contin.* **2024**, *80*, 3463–3484. [CrossRef]
3. Chen, X.; Zhang, B.; Gao, D. Bearing fault diagnosis base on multi-scale CNN and LSTM model. *J. Intell. Manuf.* **2021**, *32*, 971–987. [CrossRef]
4. Khorram, A.; Khalooei, M.; Rezghi, M. End-to-end CNN+ LSTM deep learning approach for bearing fault diagnosis. *Appl. Intell.* **2021**, *51*, 736–751. [CrossRef]
5. Huang, T.; Zhang, Q.; Tang, X.; Zhao, S.; Lu, X. A novel fault diagnosis method based on CNN and LSTM and its application in fault diagnosis for complex systems. *Artif. Intell. Rev.* **2022**, *55*, 1289–1315. [CrossRef]
6. Borré, A.; Seman, L.O.; Camponogara, E.; Stefenon, S.F.; Mariani, V.C.; Coelho, L.d.S. Machine Fault Detection Using a Hybrid CNN-LSTM Attention-Based Model. *Sensors* **2023**, *23*, 4512. [CrossRef]
7. Abboush, M.; Bamal, D.; Knieke, C.; Rausch, A. Intelligent Fault Detection and Classification Based on Hybrid Deep Learning Methods for Hardware-in-the-Loop Test of Automotive Software Systems. *Sensors* **2022**, *22*, 4066. [CrossRef]
8. Salih, A.M.; Raisi-Estabragh, Z.; Galazzo, I.B.; Radeva, P.; Petersen, S.E.; Lekadir, K.; Menegaz, G. A perspective on explainable artificial intelligence methods: SHAP and LIME. *Adv. Intell. Syst.* **2025**, *7*, 2400304. [CrossRef]
9. Brusa, E.; Cibrario, L.; Delprete, C.; Di Maggio, L.G. Explainable AI for Machine Fault Diagnosis: Understanding Features' Contribution in Machine Learning Models for Industrial Condition Monitoring. *Appl. Sci.* **2023**, *13*, 2038. [CrossRef]

10. Łuczak, D.; Brock, S.; Siembab, K. Cloud Based Fault Diagnosis by Convolutional Neural Network as Time–Frequency RGB Image Recognition of Industrial Machine Vibration with Internet of Things Connectivity. *Sensors* **2023**, *23*, 3755. [CrossRef]
11. Kolar, D.; Lisjak, D.; Pająk, M.; Pavković, D. Fault Diagnosis of Rotary Machines Using Deep Convolutional Neural Network with Wide Three Axis Vibration Signal Input. *Sensors* **2020**, *20*, 4017. [CrossRef]
12. Xu, G.; Xu, J.; Zhu, Y. LSTM-based estimation of lithium-ion battery SOH using data characteristics and spatio-temporal attention. *PLoS ONE* **2024**, *19*, e0312856. [CrossRef]
13. Zhang, X.; Sun, J.; Shang, Y.; Ren, S.; Liu, Y.; Wang, D. A novel state-of-health prediction method based on long short-term memory network with attention mechanism for lithium-ion battery. *Front. Energy Res.* **2022**, *10*, 972486. [CrossRef]
14. Yoo, Y.; Lee, C.Y.; Zhang, B.T. Multimodal anomaly detection based on deep auto-encoder for object slip perception of mobile manipulation robots. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: New York, NY, USA, 2021; pp. 11443–11449. [CrossRef]
15. Pham, T.-A.; Lee, J.-H.; Park, C.-S. MST-VAE: Multi-Scale Temporal Variational Autoencoder for Anomaly Detection in Multivariate Time Series. *Appl. Sci.* **2022**, *12*, 10078. [CrossRef]
16. Wang, R.; Dong, E.; Cheng, Z.; Liu, Z.; Jia, X. Transformer-based intelligent fault diagnosis methods of mechanical equipment: A survey. *Open Phys.* **2024**, *22*, 20240015. [CrossRef]
17. Ding, Y.; Jia, M.; Miao, Q.; Cao, Y. A novel time–frequency Transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings. *Mech. Syst. Signal Process.* **2022**, *168*, 108616. [CrossRef]
18. *ISO 26262-1:2018*; Road vehicles—Functional safety—Part 1: Vocabulary; Technical Committee: ISO/TC 22/SC 32. ISO (International Standard Organization): Geneva, Switzerland, 2018. Available online: https://www.iso.org/standard/68383.html (accessed on 10 October 2025).
19. Liu, H.; Xu, Q.; Han, X.; Wang, B.; Yi, X. Attention on the key modes: Machinery fault diagnosis transformers through variational mode decomposition. *Knowl.-Based Syst.* **2024**, *289*, 111479. [CrossRef]
20. Kumar, P. AI-driven Transformer Model for Fault Prediction in Non-Linear Dynamic Automotive System. *arXiv* **2024**, arXiv:2408.12638. [CrossRef]
21. Hou, Y.; Wang, J.; Chen, Z.; Ma, J.; Li, T. Diagnosisformer: An efficient rolling bearing fault diagnosis method based on improved Transformer. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106507. [CrossRef]
22. Lv, H.; Chen, J.; Pan, T.; Zhang, T.; Feng, Y.; Liu, S. Attention mechanism in intelligent fault diagnosis of machinery: A review of technique and application. *Measurement* **2022**, *199*, 111594. [CrossRef]
23. Liu, G.; Zhu, B. A Review of Intelligent Device Fault Diagnosis Technologies Based on Machine Vision. *arXiv* **2024**, arXiv:2412.08148. [CrossRef]
24. Yang, Y.; Tu, F.; Huang, S.; Tu, Y.; Liu, T. Research on CNN-LSTM DC power system fault diagnosis and differential protection strategy based on reinforcement learning. *Front. Energy Res.* **2023**, *11*, 1258549. [CrossRef]
25. Cui, J.; Kuang, W.; Geng, K.; Jiao, P. Intelligent fault diagnosis and operation condition monitoring of transformer based on multi-source data fusion and mining. *Sci. Rep.* **2025**, *15*, 7606. [CrossRef]
26. Zhang, W.; Yang, J.; Bo, X.; Yang, Z. A dual attention mechanism network with self-attention and frequency channel attention for intelligent diagnosis of multiple rolling bearing fault types. *Meas. Sci. Technol.* **2023**, *35*, 036112. [CrossRef]
27. Attallah, O.; Ibrahim, R.A.; Zakzouk, N.E. A lightweight deep learning framework for transformer fault diagnosis in smart grids using multiple scale CNN features. *Sci. Rep.* **2025**, *15*, 14505. [CrossRef]
28. Cao, L.; Zhang, H.; Meng, Z.; Wang, X. A parallel GRU with dual-stage attention mechanism model integrating uncertainty quantification for probabilistic RUL prediction of wind turbine bearings. *Reliab. Eng. Syst. Saf.* **2023**, *235*, 109197. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Unsupervised Learning of Energy States in Automated Storage Systems with Self-Organizing Maps

Manal Jammal *, Javier Parra Domínguez *, Laura Grande-Pérez and Fernando De la Prieta Pintado

Department of Informatics and Automatics, 37008 Salamanca, Spain
* Correspondence: mjammal@usal.es (M.J.); javierparra@usal.es (J.P.D.)

Abstract: Energy efficiency in industrial environments is subject to regulatory and economic constraints. Automated intralogistics systems, such as High Rack Storage Systems (HRSS), exhibit complex and dynamic energy patterns. This paper proposes an unsupervised learning approach that uses Self-Organizing Maps (SOMs) to characterize operational energy states from HRSS measurements (power, voltage, and position). After preprocessing, we train an SOM and apply Watershed segmentation to obtain a topological map of states, and we analyze state transitions with a Markov model to study persistence and switching behavior. The approach yields an interpretable taxonomy of energy use and highlights operating conditions associated with different efficiency levels, as well as central states that influence system behavior. While the study focuses on a single demonstrator, the results suggest that SOM can support explainable monitoring and analysis of industrial energy behavior and may help guide proactive energy-management decisions in Industry 4.0 settings.

Keywords: artificial intelligence; machine learning; Self-Organizing Maps; unsupervised learning; energy efficiency; autonomous systems

1. Introduction

The energy demand of the industrial sector represents a significant challenge in the European ecological transition. In 2022, industry accounted for approximately 25% of final energy consumption in the European Union, according to Eurostat data (Eurostat. (2024). Final energy consumption in industry—detailed statistics. European Commission). This category refers to energy used by sectors, excluding losses from transformation and transport. To address this issue, Directive (EU) 2023/1791 on energy efficiency sets a binding target of reducing final energy consumption by 11.7% by 2030 compared to reference projections, doubling previous commitments and reinforcing legal obligations (Unión Europea. (2023). Directiva (UE) 2023/1791 del Parlamento Europeo y del Consejo de 13 de septiembre de 2023 relativa a la eficiencia energética y por la que se modifica el Reglamento (UE) 2023/955. Diario Oficial de la Unión Europea). In line with this policy, the Spanish National Integrated Energy and Climate Plan 2021–2030 (PNIEC) (Gobierno de España—Ministerio para la Transición Ecológica y el Reto Demográfico (MITECO), (2020). Plan Nacional Integrado de Energía y Clima (PNIEC) 2021–2030) identifies industrial energy efficiency as one of the strategic pillars for decarbonizing the Spanish economy, highlighting the potential for improvement through intelligent energy management and automation technologies.

Industrial organizations therefore face growing pressure from investors, customers, and regulators to demonstrate commitment to reducing carbon emissions, making energy

efficiency both an economic and strategic priority. Automated storage systems illustrate this challenge well. These systems, widely used in industrial environments for inventory management and automated material handling, integrate multiple electromechanical components that operate in coordination, such as conveyor belts, motorized vertical rails, position sensors, and programmable control systems. Their complexity results in dynamic and multivariate energy consumption patterns, where overall efficiency depends not only on the performance of individual components but also on their temporal interactions. At the same time, the development of regulatory frameworks for renewable self-consumption, such as net-metering and net-billing systems implemented in various countries [1], is shaping an energy ecosystem that promotes the adoption of energy optimization solutions in industrial environments. Despite the continuous generation of large volumes of operational data through integrated sensors including power, voltage, position, and activation states, there remains a technological gap between the abundance of available data and its effective use for energy optimization [2]. Current control systems are predominantly reactive, responding only to past events, and they lack the ability to proactively adjust operating parameters to minimize energy consumption without compromising functionality. This limitation is compounded by the stochastic nature of consumption patterns, where transitions between operating states generate unpredictable peaks in energy demand [3].

The motivation for developing intelligent solutions rests on three main pillars: economic, since optimization can reduce costs by 15–30% according to recent studies [4]; regulatory, given the growing number of legal obligations and penalties for non-compliance; and environmental, as intelligent energy management systems improve efficiency while also providing strategic value. Their implementation strengthens corporate image, facilitates access to sustainable financing, and contributes to achieving the Sustainable Development Goals established in the 2030 Agenda, particularly SDGs 7, 9, and 13 (Naciones Unidas. (2015). Transformar nuestro mundo: la Agenda 2030 para el Desarrollo Sostenible. Resolución A/RES/70/1 de la Asamblea General de la ONU.). The current technological context offers favorable conditions for applying advanced artificial intelligence techniques to industrial energy optimization. The proliferation of low-cost, high-precision IoT sensors has enabled access to real-time operational data, while the exponential increase in computational capacity supports the application of complex machine learning algorithms in industrial environments [5]. The availability of machine learning frameworks such as TensorFlow and PyTorch, together with the growing volume of data generated by industrial IoT sensors, has facilitated the development of AI-based systems for energy analysis. Self-Organizing Maps (SOM), a type of neural network for unsupervised learning, allow the identification of patterns in multivariate data and the derivation of discrete events from continuous signals [6], providing an approach to segment operational states and characterize energy performance in complex systems.

The integration of these techniques into industrial energy optimization systems is the research focus of this work.

This paper is organized into six main sections. Following this introduction, Section 2 presents a systematic review of the state of the art on unsupervised learning techniques applied to energy optimization in industrial systems, with particular emphasis on Self-Organizing Maps. Section 3 details the methodology employed, including exploratory data analysis of the HRSS system, the design and training of the SOM model, and the segmentation procedures using Watershed and transition analysis with Markov models. Section 4 presents the results obtained, covering the topological structure of energy states, transition dynamics, operational cycles, and identified critical sequences. Section 5 discusses the implications of the findings, the limitations of the study, and future research directions.

Finally, Section 6 synthesizes the main conclusions and the practical value of the proposed methodology for proactive energy management in Industry 4.0 environments.

2. State of the Art

For this paper, a state-of-the-art review has been conducted focusing on the application of unsupervised learning techniques for the characterization and optimization of energy in automated industrial systems. The following key contributions and trends have been identified in the literature:

De Santana et al. [7] apply data mining techniques to improve energy efficiency in the auxiliary services of a petrochemical plant. Their study focuses on analyzing operational data to identify optimization opportunities through clustering and pattern analysis, and was implemented in a real industrial environment.

Gao et al. [8] propose a hybrid methodology combining machine learning with evolutionary algorithms for energy-efficient buffer and service rate allocation in manufacturing systems. Clustering techniques are used as part of a broader energy optimization framework, specifically applied to material handling systems.

In the context of automated storage systems, Zhou et al. [9] present multi-objective evolutionary algorithms for integrated production and maintenance scheduling with energy considerations. Their approach includes segmentation techniques for identifying operational configurations, although the primary focus is on multi-objective optimization rather than energy characterization.

Overall, the reviewed studies consistently identify unsupervised clustering techniques, particularly SOM, as viable methodologies for analyzing energy patterns in industrial systems, especially when structured prior knowledge of operational states is unavailable.

In the domain of robotics, Wang et al. [10] propose energy-efficient trajectory planning for industrial robots using parallel deep reinforcement learning. Their work emphasizes the role of optimization in reducing energy consumption during motion operations, complementing the broader objective of identifying efficient operational states through clustering techniques.

Bonci et al. [11] apply Deep Echo State Networks (DeepESN) for predictive maintenance based on energy data. Although their approach is supervised, it reinforces the relevance of energy-based anomaly detection, a task that SOM can address through unsupervised segmentation of operational states.

Ma [12] develops AI-driven decision support systems for sustainable energy management in smart cities. While the application context is urban rather than industrial, the conceptual framework for integrating multiple data sources and optimization algorithms is relevant to large-scale energy-aware systems, including industrial intralogistics.

Recent developments in industrial energy management reflect a shift toward more sophisticated control systems that integrate multiple operational objectives, including energy efficiency, productivity, and environmental sustainability. This transition from single-objective to multi-objective optimization aligns with the increasing complexity of industrial environments, where unsupervised learning techniques such as SOM can contribute to the segmentation and interpretation of operational states.

Guo et al. [13] propose energy-efficient learning algorithms for green wireless sensor networks in terminal-edge collaborative architectures. Although focused on communication networks, the underlying principles of distributed energy optimization are applicable to industrial systems where SOM can be used to cluster sensor data and identify energy-relevant patterns in decentralized environments.

Tieng et al. [14] implement an explainable AI-based facility control system designed to reduce energy consumption and carbon emissions. Their emphasis on interpretability

complements the use of SOM, which provides topological mappings of operational states that can be visualized and understood without requiring labeled data.

The reviewed literature also highlights the consolidation of artificial intelligence techniques for predictive maintenance and monitoring in industrial systems. Akyaz and Engin [15] develop a machine learning-based predictive maintenance system for artificial yarn machines, using operational data to anticipate intervention needs. While their approach is supervised, the task of identifying anomalous energy patterns is closely related to SOM's capability to detect deviations from normal operational states.

Mou et al. [16] present a machine learning approach for energy-efficient scheduling in intelligent transportation systems under dynamic conditions. Although focused on routing and scheduling, their work reinforces the importance of adaptive decision-making based on operational context, supporting the use of SOM to identify and respond to energy-critical transitions.

Cirera et al. [17] propose a data-driven optimization method for industrial refrigeration systems, integrating demand forecasting with control strategies. Their approach demonstrates the value of unsupervised segmentation in identifying consumption patterns, which can be enhanced through SOM-based clustering of operational states.

Ayadi et al. [18] contribute deep learning-based soft sensors for flexible automation in industrial environments. Their work addresses the estimation of unmeasured process variables, a challenge that SOM can support by clustering multivariate sensor data to infer latent operational conditions.

Pathak [19] explores energy savings through synergistic innovation and digitalization in the pulp and paper industry. His study identifies opportunities for technological integration that align with the use of SOM in digital industrial ecosystems, where unsupervised learning can support energy-aware automation.

Finally, Morteza et al. [20] analyze the role of Industry 4.0 technologies in promoting energy sustainability. Their interpretive structural model maps the interrelations between various energy functions, providing a conceptual foundation for integrating SOM into broader frameworks for intelligent energy management in manufacturing systems.

In summary, recent research confirms a growing reliance on artificial intelligence techniques for industrial energy optimization, with unsupervised learning methods playing a particularly relevant role when labeled data or explicit event signals are not available. SOM stand out as a recurrent approach for clustering operational states, detecting anomalies, and supporting interpretable visualizations of system behavior. However, most studies remain limited to specific industrial contexts or focus primarily on predictive maintenance rather than comprehensive energy characterization. This reveals a gap in the literature concerning the systematic application of SOM to intralogistic environments, where complex multivariate signals and dynamic interactions govern energy performance. Addressing this gap, the present work proposes an SOM-based framework to characterize operational states in automated storage systems and to provide an interpretable foundation for proactive energy management in Industry 4.0 settings (Figure 1).

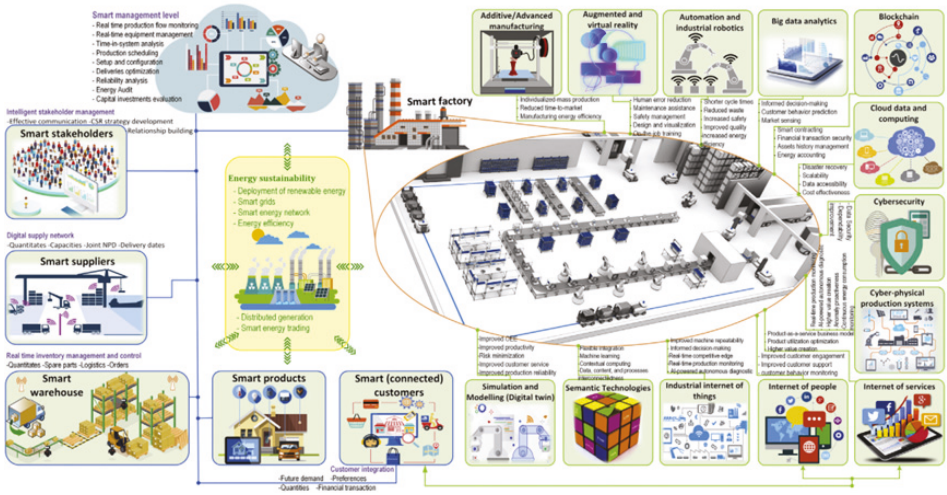


Figure 1. Schematic representation of Industry 4.0 architecture. Source: Pathak (2024).

Recent research on artificial intelligence for industrial energy management can be grouped into three main directions. First, data-driven optimization studies [7–9] apply clustering or hybrid machine-learning and evolutionary algorithms to improve process efficiency, yet these approaches focus on optimization objectives rather than the interpretable segmentation of operational states. Second, predictive-maintenance frameworks [11,15] employ deep or supervised networks to forecast failures from energy data, providing accurate prediction but limited transparency and requiring labeled anomalies. Third, explainable-AI and control-oriented systems [14,17] highlight the need for interpretable models but typically rely on explicit event markers or simplified dynamics that overlook continuous energy variability.

Across these domains, unsupervised methods particularly Self-Organizing Maps (SOMs) have been recognized for their ability to uncover latent operational patterns without prior labels. However, previous works rarely link the resulting clusters to temporal behavior or physical energy meaning. Most studies analyze static snapshots of energy consumption, whereas industrial systems operate through dynamic transitions between states that determine overall efficiency. Despite the increasing adoption of Self-Organizing Maps (SOMs) and related techniques, few studies have systematically integrated topological SOM clustering with temporal transition analysis. This work bridges that gap by introducing a hybrid SOM–Watershed and Markov-transition framework that jointly captures the spatial organization and temporal dynamics of energy states, providing an interpretable and physically grounded representation of industrial energy behavior.

3. Materials and Methods

3.1. Exploratory Data Analysis

An exploratory analysis was performed to identify components and operational cycles with the greatest potential for energy optimization. This step focused on characterizing the energy behavior of the High Rack Storage System (HRSS) under normal operating conditions, providing the foundation for subsequent unsupervised segmentation using Self-Organizing Maps (SOM).

3.1.1. Dataset Description

The study uses the HRSS dataset collected from a real automated intralogistics demonstrator developed by SmartFactoryOWL in collaboration with the Institute Industrial IT (inIT), Lemgo, Germany. The HRSS is designed as an Industry 4.0 testbed for analyzing operational efficiency and anomaly detection in automated storage and retrieval processes. Further information about the demonstrator and its architecture can be found in the Smart-

FactoryOWL website https://smartfactory-owl.de/ (accessed on 27 October 2025). The system comprises four conveyor belts (BLO, BHL, BHR, BRU) and two vertical rails (HR, HL). The central conveyors (BHL and BHR) are mounted on the rails and can move vertically, while BLO and BRU remain fixed. Each conveyor is equipped with inductive sensors for object detection and position tracking. The system is controlled by a programmable logic controller (PLC) compliant with IEC 61131 standards, which simulates typical material flow in industrial environments.

The dataset contains 25,907 time-stamped records collected under normal operating conditions, including synchronized measurements of position, power, and voltage for the six main components. Variables are grouped into four categories: (i) temporal (timestamp), (ii) positional (e.g., I_w_BLO_Weg), (iii) power (e.g., O_w_HR_power), and (iv) voltage (e.g., O_w_BHL_voltage). All signals were preprocessed through missing-value imputation and Z-score normalization to ensure consistency across features.

3.1.2. Exploratory Energy Analysis

To establish a reference for energy performance, a baseline was calculated for each component using the 25th percentile of its active power distribution. Only active states were considered, filtering out idle periods with thresholds of 100 W for conveyors and 500 W for vertical rails. This procedure ensures that the baseline reflects typical operating conditions rather than transient noise.

The comparison between baseline and mean active power revealed significant variability across components. Although BRU and BHR exhibited minimal deviation from their baselines, components such as BHL, HR, and HL showed substantial gaps, indicating a higher optimization potential. These findings motivated the application of unsupervised learning techniques to segment operational states according to their energy efficiency.

Furthermore, temporal analysis of the total power of the system highlighted frequent and abrupt variations, with an average change every 37 ms. This dynamic behavior reinforces the need for models capable of capturing multivariate patterns and transitions between operating states, a requirement addressed through SOM in the following section.

3.1.3. Exploratory Energy Analysis (PCA)

To identify dominant energy patterns and reduce dimensionality, a Principal Component Analysis (PCA) was applied to the six power variables of the HRSS components. All features were standardized using Z-score normalization to mitigate scale differences. The first three principal components explained 78% of the total variance, with PC1 and PC2 together accounting for 59.3% refer to Table 1.

The loading structure revealed two main functional clusters: (i) BHR and HR with strong positive contributions to PC1, and (ii) BHL, HL, and BLO with negative loadings, while BRU showed an independent behavior in PC2. These results indicate heterogeneous energy dynamics across components, supporting the need for unsupervised segmentation to capture operational states with distinct energy profiles. The combination of Principal Component Analysis (PCA) and Self-Organizing Maps (SOMs) was adopted to ensure a balance between interpretability and clustering performance. PCA provides a transparent linear transformation that reduces redundancy among correlated process variables while preserving their physical meaning in terms of power, voltage, and positional behavior.

Table 1. Principal component loadings for the first three axes.

Component	PC1	PC2	PC3
BLO	−0.525	−0.263	0.738
BHL	−0.631	0.392	−0.510
BHR	0.658	0.567	0.012
BRU	0.488	−0.652	−0.400
HR	0.648	0.563	0.310
HL	−0.588	0.529	−0.090

3.2. SOM Modeling

The objective of the SOM within the energy optimization framework is to generate a topological representation of the normal operating states of the HRSS, classifying each state according to its energy performance index. This representation serves three purposes: (i) establishing a reference map of normal energy behavior that enables deviation detection through distance calculations to the BMU, (ii) identifying regions of high and low energy efficiency to guide optimization strategies, and (iii) providing an automatic segmentation of operating states to support the analysis of temporal transitions.

The model was trained using power, voltage, and position variables from the six main components, excluding timestamps and labels as they do not provide relevant energy information. Preprocessing included missing-value imputation via linear interpolation, outlier detection using the interquartile range criterion (Q1 $-$ 1.5 $\times$ IQR and Q3 $+$ 1.5 $\times$ IQR), and Z-score normalization to ensure homogeneous variable scales. Additionally, all numeric variables, including power, voltage, and position signals, were concatenated into a unified feature matrix after standardization. The "position" variables were incorporated directly in their normalized numeric form without derivative computation, as the objective of this stage was to capture global energy behavior rather than detailed kinematic dynamics. These details have been explicitly included to ensure reproducibility of the preprocessing workflow. The SOM architecture was implemented as a two-dimensional rectangular grid, with dimensions determined using an adaptive heuristic based on dataset size. The number of neurons followed the empirical rule $5\sqrt{n}$, where n is the number of training samples. For 25,907 records, this yielded approximately 806 neurons, distributed in a 32 $\times$ 26 grid to maintain balanced proportions.

The weight space dimensionality was set to 18, corresponding to the number of input variables after preprocessing. Each neuron (i, j) in the grid is defined by a weight vector:

$$\mathbf{w}_{i,j} = [w^1_{i,j}, w^2_{i,j}, \ldots, w^{18}_{i,j}],$$

where each component corresponds to a specific variable. Training was carried out with the Kohonen algorithm using exponential decay for both the learning rate and the neighborhood radius. For each input vector $\mathbf{x}(t)$, the BMU is identified as:

$$c(t) = \arg\min_{i,j} \|\mathbf{x}(t) - \mathbf{w}_{i,j}(t)\|,$$

and weight updates are applied as

$$\mathbf{w}_{i,j}(t+1) = \mathbf{w}_{i,j}(t) + \alpha(t) \cdot h_{c,i,j}(t) \cdot [\mathbf{x}(t) - \mathbf{w}_{i,j}(t)],$$

where $\alpha(t)$ is the learning rate and $h_{c,i,j}(t)$ is the neighborhood function.

The neighborhood function follows a Gaussian distribution:

$$h_{c,i,j}(t) = \exp\left(-\frac{d_{c,i,j}^2}{2\sigma(t)^2}\right),$$

where $d_{c,i,j}$ is the Euclidean distance between the BMU and neuron (i, j) on the SOM grid.

The grid size and aspect ratio of the SOM were determined automatically using a heuristic proportional to the square root of the number of samples ($\approx 5\sqrt{N}$), adjusted according to the principal-component variance ratio to preserve the data topology. Training was performed with a learning rate of 0.5, an initial neighborhood radius $\sigma = 1.0$, and 1000 iterations using the Gaussian neighborhood function implemented in MiniSom. After training, the Watershed segmentation was applied to the inverted U-matrix ($-U$) to detect clusters as topological basins, where local maxima were selected as region markers.

Temporal decay of parameters was modeled with exponential functions:

$$\alpha(t) = \alpha_0 \cdot \exp\left(-\frac{t}{\tau_\alpha}\right), \quad \sigma(t) = \sigma_0 \cdot \exp\left(-\frac{t}{\tau_\sigma}\right),$$

where $\tau_\alpha = 1000$ and $\tau_\sigma = 500$ represent the time constants for decay.

The training procedure is summarized in Algorithm 1.

Algorithm 1: SOM Training Procedure

Input: $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{n \times d}$ (normalized HRSS data),
grid_size = (rows, cols), σ_0, α_0, max_iter
Output: $W = \{w_{i,j}\} \in \mathbb{R}^{\text{rows} \times \text{cols} \times d}$ (trained weight matrix)

Initialize weight matrix W randomly from uniform distribution $[-1, 1]$;
Set decay constants: $\tau_\alpha \leftarrow \frac{\text{max_iter}}{4}$, $\tau_\sigma \leftarrow \frac{\text{max_iter}}{8}$;
for $t = 1$ **to** *max_iter* **do**
    $\alpha(t) \leftarrow \alpha_0 \cdot \exp\left(-\frac{t}{\tau_\alpha}\right)$;
    $\sigma(t) \leftarrow \sigma_0 \cdot \exp\left(-\frac{t}{\tau_\sigma}\right)$;
    Select random sample $x(t)$ from X;
    // `Find BMU` **forall** *neurons* (i, j) **do**
        $\text{dist}[i, j] \leftarrow \|x(t) - w_{i,j}(t)\|_2$;
    $(c_1, c_2) \leftarrow \arg\min(\text{dist})$;
    // `Update neighborhood` **forall** *neurons* (i, j) **do**
        $d_{\text{grid}} \leftarrow \|(i, j) - (c_1, c_2)\|_2$;
        $h \leftarrow \exp\left(-\frac{d_{\text{grid}}^2}{2\sigma(t)^2}\right)$;
        $w_{i,j}(t+1) \leftarrow w_{i,j}(t) + \alpha(t) \cdot h \cdot [x(t) - w_{i,j}(t)]$;
return W;

After training, the U-matrix was computed as the average distance between each neuron and its immediate neighbors. For neuron (i, j):

$$U[i, j] = \frac{1}{|N(i, j)|} \sum_{(k,l) \in N(i,j)} \left\|\mathbf{w}_{i,j} - \mathbf{w}_{k,l}\right\|,$$

where $N(i, j)$ is the set of immediate neighbors (8-connectivity), and $\|\cdot\|$ denotes Euclidean distance.

Energy performance evaluation was based on an index relating observed consumption to the baseline established in the exploratory analysis. For each neuron, the mean energy

performance index of its assigned samples was computed. The index for each component c is defined as

$$\mathrm{IPE}_c = \frac{\mathrm{baseline}_c}{\mathrm{average_power}_c},$$

and the global index as

$$\mathrm{IPE}_{\mathrm{system}} = \frac{\sum_c (\mathrm{baseline}_c \cdot \mathrm{IPE}_c)}{\sum_c \mathrm{baseline}_c}.$$

Segmentation of the U-matrix was performed with the Watershed algorithm, treating the matrix as a topographic surface. The U-matrix was inverted before segmentation, since Watershed identifies basins as regions, whereas here the regions of interest correspond to high U-matrix values.

For each cluster k, characterization metrics were computed: number of assigned samples, average component power, standard deviation of consumption, mean energy performance index, and dominant energy state. This allowed the physical interpretation of clusters representing specific operating states of the HRSS.

Finally, a temporal transition analysis was carried out to characterize the dynamic behavior of the HRSS. Each observation was assigned to its BMU and corresponding cluster, generating a sequence of states over time. Based on this sequence, a Markov transition matrix was constructed, with rows representing current states and columns representing subsequent states, normalized to obtain conditional probabilities. Transition entropy was calculated for each state to quantify stability, where high entropy indicates multiple possible evolutions and low entropy reflects stable trajectories.

4. Results

The results are organized according to the methodological workflow, from the topological characterization of energy states with Self-Organizing Maps (SOMs) to the analysis of temporal dynamics and critical sequences.

4.1. Topological Structure and U-Matrix

Applying the Watershed algorithm to the U-matrix produced 52 initial regions, of which 48 contained sufficient sample assignments for subsequent statistical analysis and were retained as the effective clusters of the model. This automatic segmentation yields an operational taxonomy of the HRSS that captures energy variability across the six main components.

The learned topology is illustrated with three complementary views (Figure 2): (i) a 2D U-matrix, where high-dissimilarity boundaries (yellow) separate distinct operating configurations and low-dissimilarity areas (blue) group similar patterns; (ii) a 3D surface that highlights topological valleys associated with stable configurations; and (iii) an energy-efficiency map that labels each cluster according to its performance index relative to the baseline (green = near-optimal; red = significant gaps).

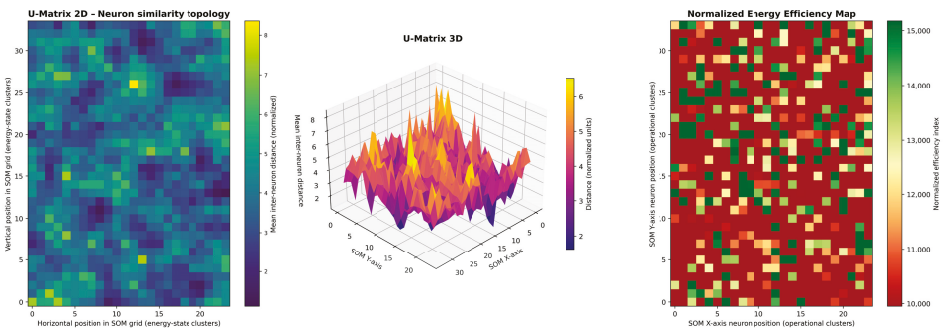


Figure 2. U-matrix and energy-efficiency map.

Overall, the SOM reveals heterogeneous operating configurations with different efficiency levels. The prevalence of medium-to-low efficiency clusters is consistent with the gaps previously observed for BHL, HR, and HL. This topology provides a basis for automatically identifying transitions toward more efficient states while preserving system functionality.

4.2. Operational Dynamics and Transitions

Once the SOM-based taxonomy of states was established, we analyzed temporal transitions among the identified configurations. The state-transition graph (Figure 3) was derived from the BMU assignments of the 25,907 time-stamped samples, producing discrete state sequences over time. The resulting graph comprises 48 nodes and 929 edges, with a connection density of 0.41. Connectivity concentrates in a central region, indicating hub states from which the system can reach multiple alternatives. Transitions are more frequent between states with similar efficiency, whereas moves between opposite categories are less common.

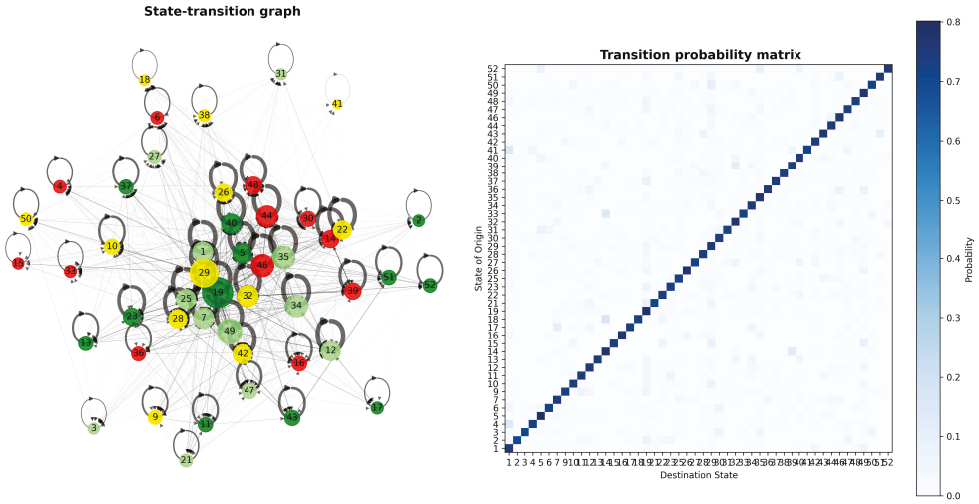


Figure 3. State-transition graph and transition probability matrix.

The transition probability matrix exhibits a dominant diagonal, with self-transition probabilities above 0.8 for most states, evidencing strong operational inertia. Probabilities for transitions between different states are mostly below 0.2, indicating that configuration changes are relatively infrequent under normal HRSS operation. Centrality analysis identifies states 29, 19, 46, 49, 25, 35, and 32 as hubs, characterized by high in- and out-degree and acting as strategic nodes for intervention.

4.3. Operational Cycles

A coefficient of variation of 71.7% indicates high inconsistency in standard transport sequence durations. Short cycles (25 samples) correspond to direct transfers that minimize waiting configurations. Long cycles (>325 samples) reflect extended stops in BHL, HR, and HL components previously identified as major contributors to energy gaps. These findings support prioritizing interventions in sequences exceeding ∼240 samples, where additional time translates into unproductive energy consumption (Table 2).

Table 2. Operational cycle characteristics of the HRSS.

Metric	Value
Complete operational cycles detected	183
Average duration (samples)	141.1
Standard deviation (samples)	101.2
Coefficient of variation	71.7%
Most frequent cycle (samples)	∼240
Minimum duration (samples)	25
Maximum duration (samples)	>325

4.4. Transition Structure, Entropy, and Hubs

State-wise transition entropy ranges from 0 to 1.2, with an average of 0.288 (Figure 4). Higher-entropy states (1, 2, 5, 29–35, 42–49) act as branching points where the system can evolve toward multiple configurations. Several of these overlap with the hub set (29, 19, 46, 49, 25, 35, 32), reinforcing their role as bifurcation nodes that influence subsequent energy demand.

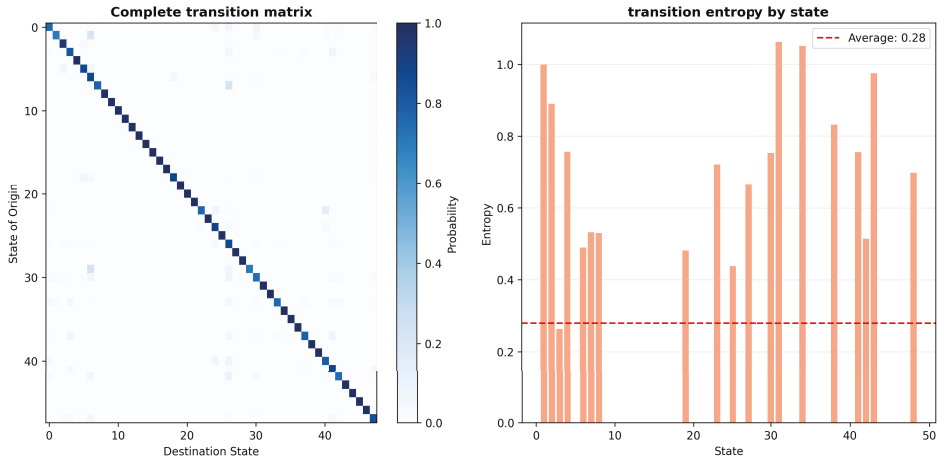


Figure 4. Complete transition matrix and transition entropy by state.

4.5. Energy Distribution by State and Critical Sequences

The transition graph filtered by probabilities >0.1 (Figure 5) shows nodes colored by energy class (dark green = OPTIMAL; light green = EFFICIENT; yellow = NORMAL; orange = SUBOPTIMAL; red = INEFFICIENT). Classes are heterogeneously distributed across the topology rather than forming cohesive clusters, suggesting that the HRSS does not follow direct preferential routes toward high-efficiency states and may require passages through less efficient states to reach optimal configurations. Each identified energy-consumption state was subsequently interpreted in relation to the physical operation of the HRSS. Clusters with low total power and short activation periods correspond to idle standby or positioning without load phases, whereas medium-efficiency states are typically associated with cargo access and walking movements. The lowest-efficiency states INEFFICIENT were characterized by prolonged activation at high power levels and irreular transition sequences, reflecting loaded lifting or suboptimal movement patterns. Conversely,OPTIMAL and EFFICIENT states coincided with balanced operation cycles, stable voltage profiles, and minimal idle durations.

The temporal evolution (Figure 6) and Temporal sequence of states (Figure 7) reveals three operational bands: (i) an upper band oscillating among OPTIMAL and EFFICIENT states (35–48), (ii) an intermediate band dominated by NORMAL states (20–30), and (iii) a lower band with transitions among OPTIMAL, EFFICIENT, and INEFFICIENT states (1–10).

Vertical transitions between bands are infrequent. The time distribution by class indicates that the system operates in EFFICIENT states for 49.8% of the time and in OPTIMAL states for 13.8%, while INEFFICIENT states account for 15.9%. This distribution implies an 398 estimated potential energy improvement of approximately 8 states were reduced through control adjustments. These values are obtained directly from 400 the transition and duration analysis implemented in the TransitionAnalyzer module.

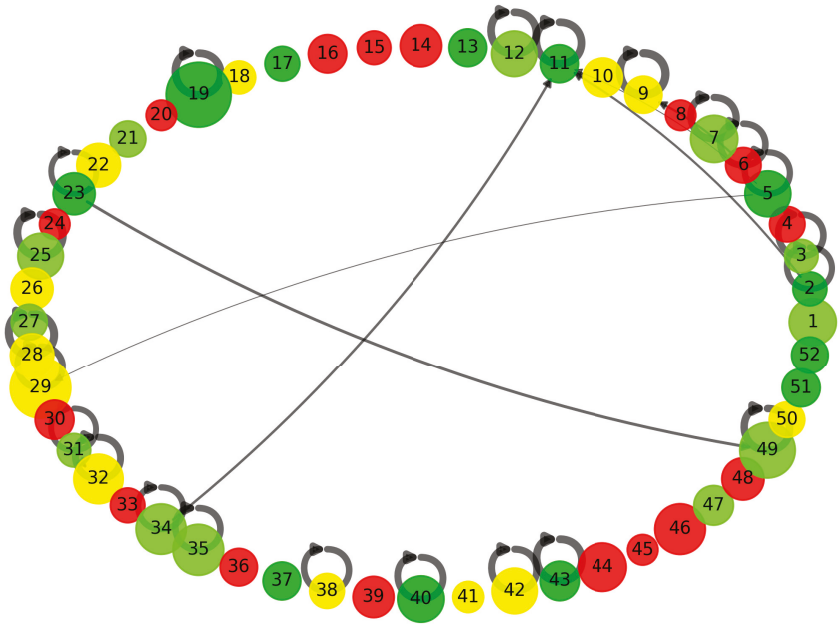


Figure 5. Transition graph with probabilities above 0.1 (nodes colored by energy class).

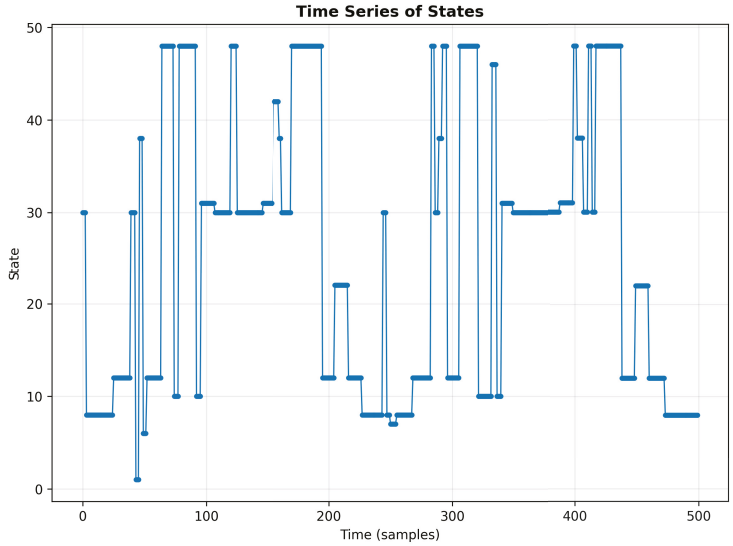


Figure 6. Temporal sequence of states (first 1000 samples).

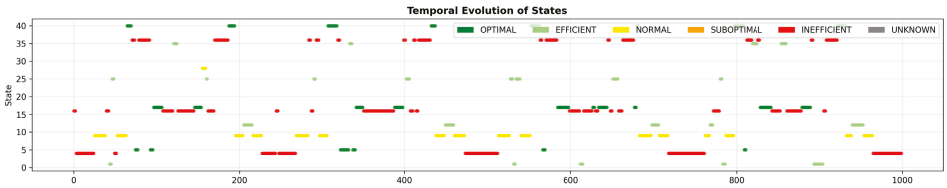


Figure 7. Temporal evolution of states and time distribution by class.

Figure 8 illustrates the most frequent transition sequences leading to inefficient states. Each horizontal bar shows the frequency of a specific sequence of operational states that most often lead to inefficient configurations Repeated patterns such as $(8, 8, 8, 8, 8)$ or $(29, 29, 29, 29, 29)$ indicate prolonged residence in a single state typically sustained motor or lift activation without effective load movement resulting in elevated power consumption with limited productive output. The most frequent involves prolonged persistence in state 6 (868 occurrences), followed by repetitions of state 48 (606 occurrences) and combinations of states 37, 48, and 7. These configurations activate multiple components without effective load movement, producing energy consumption with limited productive output. The transition subgraph among states 48, 7, 3, and 6 is particularly dense, confirming their joint contribution to energy losses.

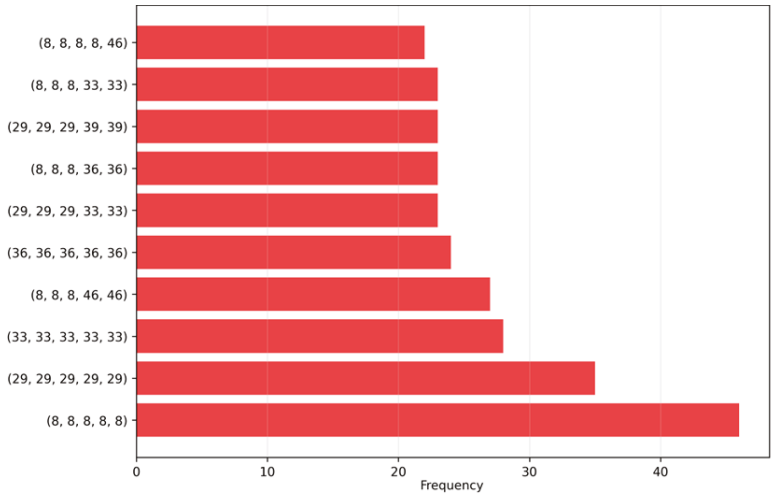


Figure 8. Top sequences leading to INEFFICIENT states.

4.6. SOM–Watershed Comparison with Other Clustering Methods

To assess the relative performance of the proposed SOM–Watershed framework, we conducted a quantitative benchmark against conventional clustering approaches, including K-Means and DBSCAN, both applied to the same normalized feature space.

Table 3 summarizes the obtained results.

The analysis revealed clear differences among the methods. DBSCAN produced a very large number of micro-clusters (over one thousand) with a sparse transition matrix, yielding low entropy values that were not meaningful due to insufficient connectivity. K-Means generated around fifty clusters with moderate stability but relatively high transition entropy, indicating fragmented state transitions. In contrast, the SOM–Watershed approach resulted in twenty-two interpretable energy states, with the highest temporal stability (average dwell time = 6.25 samples), balanced connectivity (matrix density = 0.36), and dominant transition patterns (share of $p > 0.1 = 0.06$), suggesting a more ordered and reproducible state partitioning.

Overall, these results indicate that the SOM–Watershed method achieves the most balanced trade-off between segmentation granularity, temporal coherence, and interpretability, whereas K-Means tends to over-fragment and DBSCAN severely over-segments the feature space, as summarized in Table 3.

Table 3. Comparison of clustering methods.

Method	K-Means	DBSCAN	SOM–Watershed
States	50–54	1045	22
Avg dwell	4.6	5.0	6.25
Self-P	0.77	0.80	0.80
Row entropy	0.84–0.88	0.50	0.70
Share $p > 0.1$	0.03	0.002	0.06
Density	0.19	0.002	0.36

5. Discussion

This study successfully demonstrates the viability of using unsupervised learning techniques, specifically Self-Organizing Maps (SOMs), to automatically identify energy inefficiencies within automated intralogistics systems. The methodology presented provides a repeatable framework for characterising operational states and analysing their energy performance using authentic industrial data. A key finding is that energy optimization cannot be applied uniformly across all operational states. The significant energy variability observed indicates that the system's configuration and dynamic operational behaviours have a greater impact on its energy performance than its nominal design characteristics. This highlights the particular relevance of unsupervised learning techniques for energy optimization in intralogistics and suggests their potential applicability in a wide array of other industrial domains. By overcoming the limitations of conventional reactive control strategies, this approach offers a more comprehensive understanding of system behaviour. Its unsupervised nature also reduces the reliance on expert knowledge, which makes the methodology more transferable to different industrial contexts.

The SOM-based clustering provides a physically interpretable representation of the HRSS operation. Each identified energy state corresponds to a combination of mechanical and electrical conditions such as loaded lifting, horizontal travel, idle standby, or transition between motion phases. The associated efficiency levels reflect the ratio between the measured power consumption and the baseline nominal power of each subsystem, offering a direct link between data-driven states and physical system behavior. Although the Self-Organizing Map (SOM) is a well-established and classical technique, its topology-preserving and interpretable structure remains highly valuable for applications that require transparency and reproducibility, such as industrial energy analysis. The deliberate use of a mature unsupervised method in this work prioritizes explainability over algorithmic novelty, enabling direct physical interpretation of energy states and transitions.

The proposed SOM–Markov framework could be integrated into existing industrial environments to support proactive energy management and decision-making. In practice, the trained model could interface with PLC or SCADA infrastructures to continuously map real-time sensor data to the closest energy state, enabling visualization of efficiency levels and early detection of deviations from normal operation. Moreover, by revealing high-consumption or recurrent inefficient sequences, the framework could guide engineers in prioritizing control-logic adjustments or maintenance actions that contribute to measurable energy savings. The interpretable structure of the SOM and transition graph may also facilitate the development of rule-based alarms that alert operators when the system enters a critical energy pattern. As new data become available, the model could be periodically retrained, allowing adaptive and evolving monitoring capabilities.

Despite these insights, the research has a number of limitations. The experimental validation was confined to a single system under controlled conditions, which restricts the external validity and generalisability of the findings. To address this, further empirical evaluation would be needed to extend the methodology to systems with different

configurations or more complex operational patterns. Furthermore, while the study is effective at detecting inefficiencies, it does not propose or implement corrective actions. This limits its immediate practical application and creates a clear direction for future work. Another limitation of this study is the limited availability of suitable industrial datasets. Energy-focused operational data from real production systems are rarely accessible, which constrains large-scale validation and benchmarking.

Building on these foundations, future research could focus on validating the methodology across a wider range of automated storage systems to assess its robustness and generalisability. Longitudinal studies in real-world operational settings would be invaluable for evaluating the temporal stability of the identified energy patterns and measuring the effectiveness of any optimization measures that are implemented. A crucial next step is integrating this diagnostic approach with real-time control systems, transitioning from simple diagnosis to active operational optimization. Research in this area could also explore issues of reliability, safety, and graceful degradation under fault conditions.

Future developments of this framework will investigate complementary feature-engineering strategies and hybrid extensions that combine SOM interpretability with modern deep-representation learning, aiming to enhance scalability and generalization. Ultimately, combining this methodology with broader industrial self-consumption and renewable energy strategies could unlock new opportunities for holistic energy optimization within Industry 4.0 environments, including a detailed scalability assessment of the algorithm across increasing input dimensions and sampling frequencies.

6. Conclusions

This work makes three key contributions to interpretable industrial energy analytics. First, it introduces a hybrid SOM–Watershed and transition-graph framework that jointly captures the spatial and temporal behaviour of the HRSS energy system. Second, it proposes a component-level efficiency mapping approach that exposes localized performance differences across subsystems. Third, it establishes an interpretable symbolic representation of operational states directly linked to measurable physical machine behaviour. The integration of Self-Organizing Maps (SOMs) with energy performance metrics provides a practical approach for segmenting operational states and identifying patterns linked to inefficiency. This method supports a more comprehensive understanding of system behaviour and overcomes the limitations of reactive control strategies often used in industrial environments.

Author Contributions: Conceptualization, M.J., J.P.D., F.D.l.P.P. and L.G.-P.; methodology, M.J.; writing—original draft preparation, M.J. and J.P.D.; writing—review and editing, M.J. and J.P.D.; supervision, J.P.D., L.G.-P. and F.D.l.P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: This research is part of the International Chair Project on Trustworthy Artificial Intelligence and Demographic Challenge within the National Strategy for Artificial Intelligence (ENIA), in the framework of the European Recovery, Transformation and Resilience Plan. Referencia: TSI-100933-2023-0001. This project is funded by the Secretary of State for Digitalization and Artificial Intelligence and by the European Union(Next Generation).DEMIA.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ordóñez, Á.; Sánchez, E.; Rozas, L.; García, R.; Parra-Domínguez, J. Net-metering and net-billing in photovoltaic self-consumption: The cases of Ecuador and Spain. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102434. [CrossRef]
2. Wang, W.; Yang, H.; Zhang, Y.; Xu, J. IoT-enabled real-time energy efficiency optimisation method for energy-intensive manufacturing enterprises. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 362–379. [CrossRef]
3. Cao, Z.; Han, X.; Lyons, W.; O'Rourke, F. Energy management optimisation using a combined Long Short-Term Memory recurrent neural network—Particle Swarm Optimisation model. *J. Clean. Prod.* **2021**, *326*, 129246. [CrossRef]
4. Liu, J.; Qian, Y.; Yang, Y.; Yang, Z. Can Artificial Intelligence Improve the Energy Efficiency of Manufacturing Companies? Evidence from China. *Int. J. Environ. Res. Public Health* **2022**, *19*, 2091. [CrossRef] [PubMed]
5. Alex, B.; Johnson, M. A Framework for IoT-Enabled Smart Manufacturing for Energy and Resource Optimization. *arXiv* **2025**, arXiv:2502.03040.
6. Birgelen, A.V.; Niggemann, O. Using self-organizing maps to learn hybrid timed automata in absence of discrete events. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Limassol, Cyprus, 12–15 September 2017; pp. 1–8. [CrossRef]
7. de Santana, D.M.; Lourenco, S.R.; Cassiano, D.A. Data mining approach for energy efficiency improvements in a utilities supply on a petrochemical plant. *Evol. Syst.* **2023**, *14*, 1071–1081. [CrossRef]
8. Gao, S.X.; Liu, H.; Ota, J. Energy-efficient buffer and service rate allocation in manufacturing systems using hybrid machine learning and evolutionary algorithms. *Adv. Manuf.* **2024**, *12*, 227–251. [CrossRef]
9. Zhou, B.; Li, X.; Liu, W. Hybrid multi-objective opposite-learning evolutionary algorithm for integrated production and maintenance scheduling with energy consideration. *Neural Comput. Appl.* **2021**, *33*, 1587–1605. [CrossRef]
10. Wang, X.; Cao, J.; Cao, Y.; Zou, F. Energy-efficient trajectory planning for a class of industrial robots using parallel deep reinforcement learning. *Nonlinear Dyn.* **2025**, *113*, 8491–8511. [CrossRef]
11. Bonci, A.; Fredianelli, L.; Kermenov, R.; Longarini, L.; Longhi, S.; Pompei, G.; Prist, M.; Verdini, C. DeepESN Neural Networks for Industrial Predictive Maintenance through Anomaly Detection from Production Energy Data. *Appl. Sci.* **2024**, *14*, 8686. [CrossRef]
12. Ma, N. Artificial Intelligence-Driven Decision Support Systems for Sustainable Energy Management in Smart Cities. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 523–529. [CrossRef]
13. Guo, H.; Wu, R.; Ma, Y.; Li, Y.; Li, S.; Qi, B.; Sun, Y. Learning-Based Terminal-Edge Collaborative Energy-Efficient Routing Algorithm for Green RWSN. *IEEE Trans. Ind. Inform.* **2024**, *20*, 9037–9047. [CrossRef]
14. Tieng, H.; Lai, C.Y.; Fan, S.X.; Wu, T.Q. Explainable AI-based facility control system for energy saving and carbon reduction. *J. Mech. Sci. Technol.* **2025**, *39*, 2301–2310. [CrossRef]
15. Akyaz, T.; Engin, D. Machine Learning-Based Predictive Maintenance System for Artificial Yarn Machines. *IEEE Access* **2024**, *12*, 125446–125461. [CrossRef]
16. Mou, J.; Gao, K.; Duan, P.; Li, J.; Garg, A.; Sharma, R. A Machine Learning Approach for Energy-Efficient Intelligent Transportation Scheduling Problem in a Real-World Dynamic Circumstances. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 15527–15539. [CrossRef]
17. Cirera, J.; Carino, J.A.; Zurita, D.; Ortega, J.A. A data-driven-based industrial refrigeration optimization method considering demand forecasting. *Processes* **2020**, *8*, 617. [CrossRef]
18. Ayadi, R.; El-Aziz, R.M.A.; Taloba, A.I.; Aljuaid, H.; Hamed, N.O.; Khder, M.A. Deep Learning-Based Soft Sensors for Improving the Flexibility for Automation of Industry. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5450473. [CrossRef]
19. Pathak, C. saving energy through synergistic innovation with digitalization. *IPPTA Q. J. Indian Pulp Pap. Tech. Assoc.* **2024**, *36*, 145–148.
20. Ghobakhloo, M.; Fathi, M. Industry 4.0 and opportunities for energy sustainability. *J. Clean. Prod.* **2021**, *295*, 126427. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Machine Learning-Based State-of-Charge Prediction for Electric Bus Fleet: A Critical Analysis

Simone Volturno, Andrea Di Martino * and Michela Longo

Department of Energy, Politecnico di Milano, 20156 Milan, Italy; simone.volturno@mail.polimi.it (S.V.); michela.longo@polimi.it (M.L.)

* Correspondence: andrea.dimartino@polimi.it

Abstract: The transportation sector is undergoing a rapid energy transition. Electric Vehicles (EVs) are gradually replacing conventional ones in many different sectors, but battery management still represents a critical limitation of this process. Consequently, research in this area is expanding, aiming to develop solutions that enhance performance while minimizing environmental impact. This study addresses the application of Machine Learning (ML) techniques to estimate the battery State of Charge (SoC) for a full-electric bus fleet operating public service. The methodology is built based on the available driving data disclosed from the fleet monitoring system. The ML methods are assessed starting from model-based (MB) observers assumed as reference and performances are compared upon this basis. The datasets are retrieved from a public repository or derived from real cases, particularly referring to an electric bus fleet operating for an urban public service. The most critical limitation is the absence of the electrical input data coming from the battery, typically required by model-based approaches. Despite this, the proposed ML model achieved sufficient accuracy levels (RMSE < 0.3%) comparable to those of traditional observers. These outcomes demonstrate the potential of data-driven approaches to provide scalable and more straightforward tools for battery monitoring.

Keywords: end of life; battery; machine learning; artificial intelligence; electric vehicles

1. Introduction

The recent issue regarding global pollution has led the transportation industry to explore innovative technologies to increase sustainability. Concerns are motivated by the 20% of annual CO_2 emissions by this sector, with almost 94% of vehicles involved in the sector relying on petrol fuels [1,2]. In this context, Electric Vehicles (EVs) emerge as one of the most widely adopted alternatives, thanks to electric drives, which offer several benefits compared to traditional Internal Combustion Engines (ICEs) such as energy efficiency and regenerative braking [3]. To mitigate pollution,Hybrid Electric Vehicles (HEVs) and Battery Electric Vehicles (BEVs) have increasingly been adopted as alternative means of transportation capable of reducing global emissions. Within the private sector, sales of HEVs and BEVs increased by +26.5% and +34.5%, respectively, in 2023, compared to the previous year [4]. Regarding the public transport sector, several Italian cities—including Milan and Turin—are gradually converting their entire urban bus fleet to Electric Vehicles [5, 6]. In the road freight sector, although several companies manufacture fully electric trucks, their widespread adoption remains limited due to range constraints and lack of a dedicated charging infrastructure. However, EVs still face significant drawbacks, mainly related to the battery, the most critical component. A key issue lies in the environmental footprint

of battery production, which involves energy-intensive processes mainly related to the mining of raw materials. These practices contribute notably to pollution and raise critical questions about the long-term sustainability of mass battery manufacturing. Furthermore, the current state of battery technology limits the overall performance of EVs. Common challenges include limited driving range, high charging times and non-negligible battery weight [7]. As a result, ongoing research efforts are focusing on multiple aspects of batteries to be developed, with the aim of boosting efficiency, reliability, and sustainability. One major limiting factor is the charging infrastructure. Researchers and stakeholders are cooperating to increase the accessibility of charging stations and refine charging methods to reduce wait times. To address this issue, many studies are exploring solutions such as dynamic wireless charging, which aims to extend the driving range while reducing or eliminating charging time [8]. Inductive charging has emerged as a promising leading technology, addressing many of the drawbacks associated with conductive charging: by enclosing all electronics, it reduces metallic corrosion and enhances safety by eliminating the need to plug and unplug high-voltage connectors [9]. In particular, this procedure allows batteries to be charged during driving, thus extending the range [10,11]. In parallel, advancements in battery chemistry are being investigated to enhance energy density and minimize both size and weight. Lithium-based batteries have replaced older technologies due to their superior characteristics in durability [12]. However, several emerging battery technologies, though still in an early stage, are showing promising potential. Equally important is battery management. A rapid and accurate estimation of key parameters is essential for maximizing energy utilization, decreasing maintenance needs, and preventing or anticipating failures [13]. These efforts collectively contribute to improved performance and longer battery lifespan [14].

One of the most critical aspects when dealing with batteries is the estimation of the State of Charge (SoC) [15]. The process is accurate through measuring electrical quantities such as voltage and currents. However, in some cases, these quantities may not be monitored in real time and the estimation must be carried out in different ways. For what concerns the remote management of a full-electric vehicle fleet, the SoC must be estimated at the end of the daily service in absence of electrical data and based on kinematics and driving data only. This paper aims to present and discuss different methodologies to estimate the SoC of Li-ion batteries applied to an electric bus fleet operating public transportation service. The available dataset extracted from real operations constitutes both the basis of this work and also a limitation in the number of variables disclosed. To overcome this gap, this research implemented both traditional model-based (MB) algorithms and different Machine Learning (ML) techniques to provide a solid benchmark for the assessment of the outcome. In the first phase, a publicly released experimental electrical dataset was exploited to assess the results produced by MB observers. In the second phase, the ML algorithms are evaluated as trained on real-world data collected from the real bus fleet itself. Strengths of the presented research can be summarized as follows:

- The capability of MB observers to provide a reliable outcome with reduced margins of error is positively evaluated if reduced complexity in the descriptive part is targeted.
- The ML techniques are considered for their versatility and fast adaptability to produce output results.
- The MB observer is then selected as a benchmark to evaluate the performances of different ML algorithms selected and tested.

Conversely, limitations of this study are represented by the following:

- The lack of detailed electrical quantities in the real-world dataset extracted represents the weakest aspect for a robust outcome in ML algorithms.

- Assumptions on the theoretical charging–discharging cycles—the so-called $C-rate$ of the battery—and the driving data being aggregated for different drivers operating on each line.
- The clusterization of data for each line, based on the different topographic characteristics of the routes.
- Several ML techniques were affected by overfitting, decreasing the performances of the final estimation of the result.

This paper introduces the methodology adopted to deal with the topic in Section 2, with a deep insight into the different methods selected. An MB algorithm is built up to constitute the reference case, starting from a public electric dataset. In parallel, different ML algorithms are compared based on the actual dataset available. Adjustments were needed to correct the estimated outcome from ML algorithms, while the results from real datasets are discussed in Section 3.

2. Methodology

The presented research focuses on improving the SoC estimation of Li-ion batteries, specifically applied to an electric bus fleet. The need for a reliable SoC estimation is motivated by maintenance scheduling, which must comply with strict constraints on the residual battery life. The SoC represents the amount of energy currently stored and available in a battery. Unlike directly measurable quantities such as voltage and current, the SoC cannot be determined by direct observation; therefore, suitable techniques are required to evaluate it accurately [16]. These methods can be classified as follows:

- Direct measurement approach: The simplest technique, which relies on easily accessible signals such as voltage or current, is very sensitive to external disturbances and is generally reliable only under controlled experimental conditions [14,17].
- Model-based (MB) techniques: As the most commonly used methods, they depend on detailed mathematical models capturing the non-linear behavior of the battery. In particular, they can be classified based on the type of model involved:
  - Electro-chemical, which involves the model of electro-chemical processes or electro-thermal coupling in realizing discharging currents [18,19].
  - Equivalent Circuit Model (ECM), which implements a simplified electrical circuit model to deploy the electrical characteristics of the battery [20,21].

  Although very effective, they are quite limited by the complexity of the models involved (often electro-chemical) and the need for numerous parameters to be determined [22–24].
- Machine Learning (ML) methods: Recently, these methods have gained interest due to their ease of implementation and adaptability, being capable of identifying complex non-linear patterns from historical data, without the need for explicit physical modeling [25–27].

This work involved the implementation and evaluation of both traditional MB algorithms and different ML models, tested with a experimental dataset to provide a solid benchmark for assessing the performance of the estimation. Two different experimental datasets were selected:

- The electrical dataset exploited for MB observers in the first phase, constituted by the incremental current open-circuit voltage (OCV) test, the Dynamic Stress Test (DST) and the Federal Urban Driving Schedule (FUDS) provided by the Center for Advanced Life Cycle Engineering (CALCE, University of Maryland, USA) [28].
- A real-world dataset composed by operative quantities related to the normal service operations collected from the bus fleet in a given time window.

2.1. Implementation of Traditional Methods

In order to choose a useful MB observer, an experimental study conducted by Zou et al. has been considered [29]. It compares four methods for SoC estimation:

- The Luenberger observer [22];
- The Kalman filter [30];
- The Proportional–Integral (PI) observer [31];
- The sliding-mode observer [32].

These algorithms are implemented and tested on a 20Ah EIG Li-ion battery, modeled using a first-order RC equivalent circuit (a Thevenin model). To assess their performance, the Urban Dynamometer Driving Schedule (UDDS) current profile, representing the power demand of an EV during a standardized driving cycle, is exploited. The evaluation focuses on three key aspects:

(a) Estimation accuracy;
(b) Rise time;
(c) Computational costs.

The results show that all four methods are capable of estimating the SoC and correcting initial estimation errors. For this last aspect, both the sliding-mode and Luenberger observers demonstrate high sensitivity to external disturbances [33]. Moreover, the sliding-mode observer exhibits a wider dispersion of errors, while the remaining three approaches yield more concentrated distributions. Notably, the PI and Luenberger observers depict error distributions with peaks higher than 0.5 at zero value, indicating a high level of accuracy in most of their estimations.

Rising time is defined as the time required for the estimation to reach convergence to the true SoC value. This reflects the responsiveness of each algorithm, with the PI observer performing best for the lowest rising time [34].

Regarding computational efficiency, an important consideration for real-time EV applications, the Luenberger and sliding-mode observers prove to be the most efficient, followed by the PI observer. Taking into account the relevance of disturbance sensitivity in EV contexts, the Kalman filter and the PI observer emerge as the most suitable solutions [35]. Due to its better error distribution, faster response, and low computational cost, the PI observer is selected in this research as a reference model to benchmark with the most promising ML-based estimation method.

The PI observer shows rapid and accurate convergence in the estimated SoC value to its true value: this allows the use of a first-order RC model within the PI observer to effectively represent the dynamic behavior of the battery [36]. As depicted in Figure 1, the equivalent circuit consists of a voltage source dependent on the SoC value ($E_0(z)$), representing the OCV of the battery, a series resistor (R_1) and an RC circuit formed by a parallel resistor (R_2) and a parallel capacitor (C_2). The electrical parameters related to the equivalent circuit were estimated through an iterative process involving the Least Square Regression method (LSQR) implemented in MATLAB R2024a. After we retrieved all the necessary data, a mathematical model of the battery was implemented in Simulink. A PI controller block was added and tuned using *pidTuner*, a built-in MATLAB function. Figure 2 represents the final model implemented in Simulink.

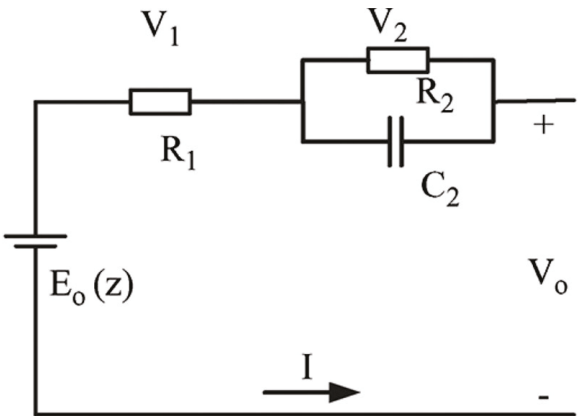


Figure 1. Battery equivalent electric circuit model (ECM).

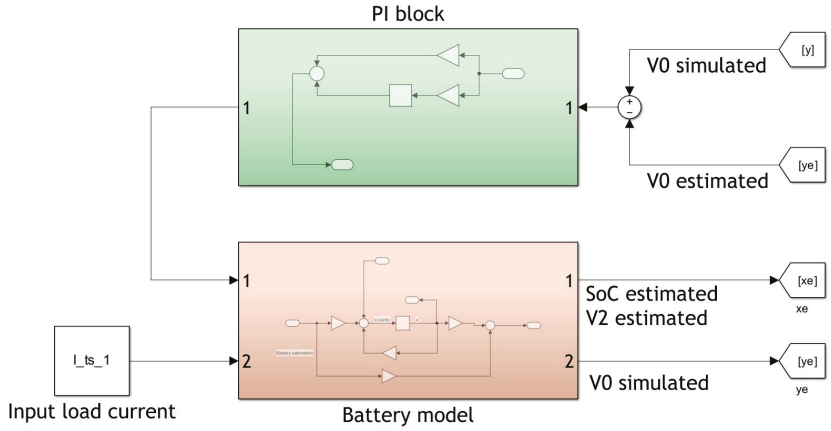


Figure 2. Simulink model of the PI observer.

To minimize error build-up during the estimation process, an anti-windup feedback loop was introduced with an experimental coefficient. This was obtained through an iterative tuning process during experimental tests. In particular, different adjustments were performed until an optimal coefficient was identified, ensuring a good compromise between fast response and limited overshoot. After the definition of all necessary parameters and the construction of the curve representing the relationship between the OCV and SoC, the implemented model was tested with two main datasets, the incremental current OCV test and the DST, both performed on a 2Ah LiNiMnCo/Graphite battery [28]. For determining the OCV-SoC curves, the incremental current OCV test was exploited: about ten reference points were collected at three different temperatures during the relaxation pauses in order to measure the OCV related to the actual SoC both when charging the battery and when discharging it. The measurements were interpolated and averaged to obtain a single relationship curve. Figure 3 illustrates the different curves obtained for specific temperature values.

Both datasets include various measured parameters such as the following:

- Time in [s];
- Load current in [A];
- Battery voltage in [V];
- External temperature in [°C];
- State of Charge (SoC) in [%].

The incremental current OCV test involves repeated flows of charging–discharging pulsed currents, while the the DST involves cycling the battery at different $C - rate$, as defined in (1),

$$C_{rate} = \frac{I[A]}{Q_{batt}[Ah]} \tag{1}$$

where I is the load current measured in [A] and Q_{batt} is the battery capacity in [Ah]. The two applied profiles differ significantly both in amplitude and nature of the stress induced. In addition, the experimental data were collected under completely different environmental conditions: the OCV test was conducted at 0 °C, while the DST was performed at 45 °C. Despite these contrasting scenarios, the difference between the estimated and actual SoC remains minimal in both cases, with a maximum deviation of less than 0.5%.

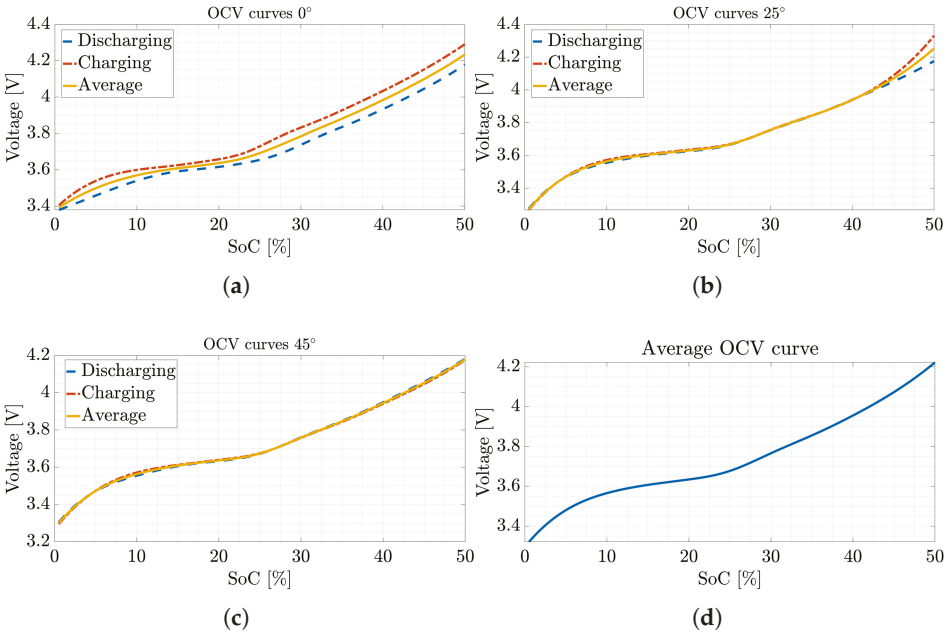


Figure 3. OCV-SoC curves for 0 °C (**a**), 25 °C (**b**), 45 °C (**c**) and averaged (**d**).

2.2. Implementation of Machine Learning Methods

Following the development of the PI observer, an alternative approach based on ML algorithms was investigated for SoC estimation. As aforementioned, training and validation phases were initially implemented for the experimental dataset, the DST and the Federal Urban Driving Schedule (FUDS), both performed at 0 °C, 25 °C and 45 °C on a 2Ah LiNiMnCo/Graphite battery [28]. The parameters contained in these datasets are organized as follows:

- Predictors, such as
  - Time in [s];
  - Load current in [A];
  - Battery voltage in [V],
  - Environmental temperature in [°C].
- A response variable (SoC).

The training and validation processes were carried out using MATLAB's built-in *Regression Learner* tool. The preliminary evaluation was carried out using various machine learning models and hyperparameter settings. Decision Trees, Ensemble Trees, and Neural Networks showed the best performance for this type of training and were therefore chosen as the primary methods for further evaluation to efficiently generate results. The tunable hyperparameters included

- For the Decision Tree models, the minimum leaf size and the surrogate decision splits.

- For the Ensemble Tree models, the minimum leaf size, the number of learners and, specifically for the Ensemble Boosted Tree, the learning rate.
- For the Neural Network models, the number of fully connected layers, the size of the layers and the activation function.

Model performances were assessed according to the Root Mean Square Error (RMSE), which quantifies the average deviation between predicted and actual values; lower RMSE values (ideally tending to zero) indicate higher prediction accuracy [37]. Figure 4 illustrates the model's accuracy obtained through training with the DST dataset after an optimization process, whose model numbers are delineated according to Table 1: as can be seen, model 2.17, representing the Ensemble Bagged Tree method, obtained the lowest RMSE value. Tables 2–4 collect the different configurations adopted for each ML algorithm, respectively.

Table 1. Types of ML models tested.

Model Number	ML Model	Type
2.1	Linear Regression	Linear
2.2	Linear Regression	Interaction linear
2.4	Linear Regression	Stepwise linear
2.5	Tree	Fine
2.6	Tree	Medium
2.7	Tree	Coarse
2.10	SVM	Cubic
2.11	SVM	Fine Gaussian
2.12	SVM	Medium Gaussian
2.13	SVM	Coarse Gaussian
2.16	Ensemble	Boosted tree
2.17	SVM	Bagged tree
2.18	Gaussian Process Regression	Squared exponential
2.19	Gaussian Process Regression	Matern 5/2
2.20	Gaussian Process Regression	Exponential
2.21	Gaussian Process Regression	Rational quadratic
2.22	Neural Network	Narrow-layered
2.23	Neural Network	Medium-layered
2.24	Neural Network	Wide-layered
2.25	Neural Network	Bilayered
2.26	Neural Network	Trilayered
2.28	Kernel	Least square regression

Table 2. Hyperparameter selection for Decision Tree models.

Model Preset	Min. Leaf Size	Surrogate Decision Splits
Fine Tree	4	Off
Medium Tree	12	Off
Coarse Tree	36	Off

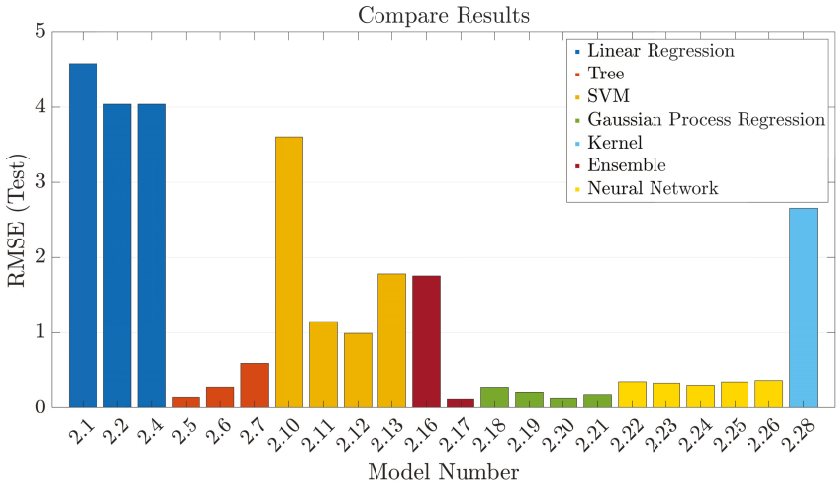


Figure 4. Regression Learner model accuracy (DST).

Table 3. Hyperparameter selection for Ensemble Tree models.

Model Preset	Min. Leaf Size	Learners	Learning Rate
Boosted	8	30	0.1
Bagged	12	30	-

Table 4. Hyperparameter selection for Neural Network models.

Model Preset	No. of Layers	Layer Size	Activation Function
Narrow	1	10	ReLU
Medium	1	25	ReLU
Wide	1	100	ReLU
Bilayered	2	10	ReLU
Trilayered	3	10	ReLU

As Figure 5 testifies, a maximum deviation of 0.6% is observed, making these results comparable to those produced by the PI observer. Similarly, the FUDS dataset also yielded very good results, with the Ensemble Bagged Tree model demonstrating the highest level of accuracy. Decision Trees, Ensemble Methods, and Neural Networks exhibited the highest levels of accuracy during the preliminary training. As a result, these algorithms were selected as the most suitable candidates for further evaluation to provide fast results.

Once the effectiveness of the ML model trained on electrical data was confirmed, the next step is represented by the development of algorithms designed to be trained on a real-world dataset. In this case, the original dataset of an electric bus fleet was acquired during the public service periods and composed only by

- Time in [s],
- Vehicle speed in [km/h] and
- Instantaneous SoC.

The bus fleet consisted of 13 EVs, categorized into two size groups:

- Small-sized buses: 8-m long vehicles each equipped with a 268.7 kWh battery pack;
- Medium-sized buses: 10-m long vehicles each equipped with a 383.4 kWh battery pack.

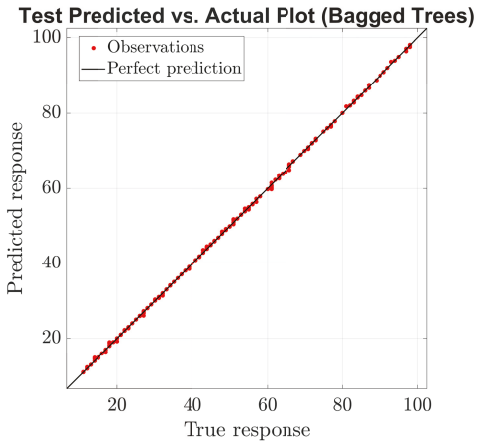


Figure 5. Prediction plot Ensemble Bagged Tree method (DST).

Although the fleet operates on seven different routes, the analysis focused on three specific lines. These were selected based on their high passenger volumes served and the distinctive topography of each route. To ensure data consistency and comparability, separate training datasets were prepared for each bus category. Each dataset was divided into allocating 50% of the samples for training and validation and the remaining 50% for testing. Moreover, the cross-validation technique was exploited. Due to the limited number of available input variables, model accuracy was enhanced through an iterative optimization process, involving both parameter tuning and dataset pre-processing. This helped to identify the most effective configuration for each model.

2.2.1. Whole Dataset

The initial stage of the analysis involved the training of several ML models using unprocessed data collected among all scheduled daily operations for each bus route. As noted earlier, the only input features available here were vehicle speed and the elapsed time since the acquisition started. Each bus line was imported individually into MATLAB, allowing the models to be trained separately for each specific route. This is due to the different topography of each line. Figure 6 displays the prediction performance for line L01C. Due to the large volume of data, the training phase proved to be computationally demanding, and in some cases, Neural Network models were unable to complete the process. A similar issue arose with line L02C operated by medium-sized vehicles, while better training results were obtained for lines L02C and L79B operated by small-sized vehicles. Although the current results are not highly accurate, they establish a foundation on which further model improvements can be built through future refinements.

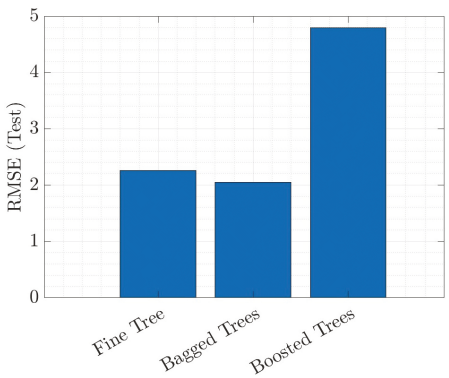


Figure 6. Regression Learner model accuracy (L01C).

The most important challenge encountered when working with the full dataset was its significant size ($\approx GB$), which often led to a slow training time or, in some cases, to the complete failure of the learning process. In addition, the relatively low training accuracy was partly attributed to the granularity of the SoC values, which resulted in statistical imbalances throughout the training phase. To overcome these limitations, a data pre-processing stage was introduced with the goal of reducing the dataset size and improving quality. Electric buses often remain idle for prolonged periods, leading to the generation of large volumes of redundant data at zero speed. To eliminate this inefficiency, all corresponding records were excluded from the dataset. A statistical analysis was then performed to assess the effects of this cleaning step. SoC values were rounded to the nearest integer and grouped into discrete "classes." The total duration associated with each class was calculated to observe the distribution. As illustrated in Figure 7, the initial distribution was highly irregular, with numerous sharp peaks. After removing zero-speed data, the class distribution became significantly more uniform and balanced. This cleaning process also resulted in a considerable reduction in the size of data, enabling a faster and more effective training phase of ML models. The refined dataset was subsequently used to train and evaluate various ML algorithms.

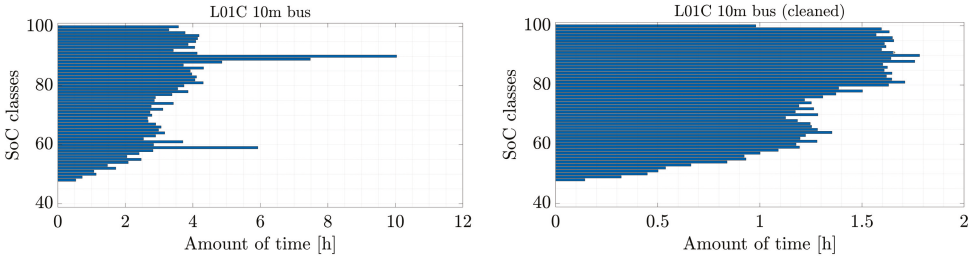


Figure 7. Original dataset distribution—cleaned dataset distribution.

Given the unsatisfactory levels of accuracy achieved for certain bus lines, a different strategy was investigated—i.e., isolating data corresponding to individual operational days. In this setting, the ML models were trained and validated using measurements from a single selected day service, while the testing phase relied on data from a different day. Although this method yielded very high validation accuracies, the test results were worse than the validation ones. This outcome is a clear manifestation of overfitting—i.e., the model adapts exceptionally well to the specific patterns of one working day but struggles to generalize when exposed to unseen data from another day. The results of two representative tests conducted with this approach are reported in Table 5.

Table 5. Best accuracy summary.

Line	Val. RMSE	Test RMSE
L02C 10m	0.09848	0.99129
L02C 8m	0.05555	1.4655

Various strategies were tested in an effort to mitigate this overfitting issue:

(a) The first step involved a cleaned dataset with a more homogeneous data distribution; however, this did not produce any notable improvements in accuracy.
(b) Next, the number of cross-validation folds within the *Regression Learner* was increased in an attempt to strengthen the model's generalization capability, but the results remained essentially unchanged.
(c) Lastly, testing subsets that incorporated data from multiple working days were assembled, but this approach also failed to produce any meaningful improvement.

Since none of these steps produced the intended outcome, this approach was ultimately discarded.

2.2.2. Whole Dataset with Additional Information

Although the previous adjustments led to some improvement, the overall model performances were still far from being optimal. To further improve predictive accuracy, additional variables were incorporated into the complete dataset. Specifically, the instantaneous acceleration profile of each vehicle was numerically derived. This choice was driven by the observation that acceleration dynamics can have a considerable impact on energy absorption and, hence, on the battery's SoC. The acceleration profile was computed for each subset of route and integrated into the cleaned dataset as an additional predictor. This enriched dataset was then used for model training in the *Regression Learner*. The inclusion of this feature strengthened the link between the predictors and the targeted variable and resulted in a consistent, albeit modest, improvement in accuracy across all routes considered, as reported in Table 6.

Another set of features incorporated into the dataset regarded historical weather conditions. For each operating day, the corresponding meteorological records from historical databases were consulted and extracted [38]. In particular, the daily average ambient temperature and wind speed were obtained and appended to the set of predictors to enhance the training of the ML models. The final group of input variables now includes

- Elapsed time [s];
- Vehicle speed [km/h];
- Vehicle acceleration [m/s^2];
- Mean external ambient temperature [°C];
- Mean wind speed [km/h].

With these parameters, the RMSE values for all lines dropped to near-zero levels, demonstrating both the high accuracy of the predictions and the strong generalization capability of the developed approach.

Throughout the analysis, the Ensemble Bagged Tree approach consistently delivered the highest performance, as testified by Figure 8, which illustrates the progression of the accuracy obtained during the various optimization stages. All routes gained substantial accuracy gains, with the most pronounced improvements observed for those lines initially featured by lower scores. Furthermore, the results reveal that the optimization not only increased accuracy but also reduced variability among the lines, leading to more uniform outcomes. Although the final RMSE values were particularly low, the prediction plots in Figure 9 indicate a maximum deviation of 0.6%, apart from a limited number of outliers. The application of such methods on the clustered dataset according to different bus lines is reported in Figure 10, with the corresponding numerical values summarized in Table 6.

Table 6. Best testing accuracy summary [RMSE].

	L01C	**L02C 10m**	**L02C 8m**	**L79B**
Whole dataset	2.0502	0.89979	0.13472	3.0099
Cleaned dataset	1.6662	0.73546	0.13629	0.60018
Acceleration dataset	1.1161	0.57974	0.10299	0.45739
Weather dataset	0.28177	0.16704	0.04861	0.11049

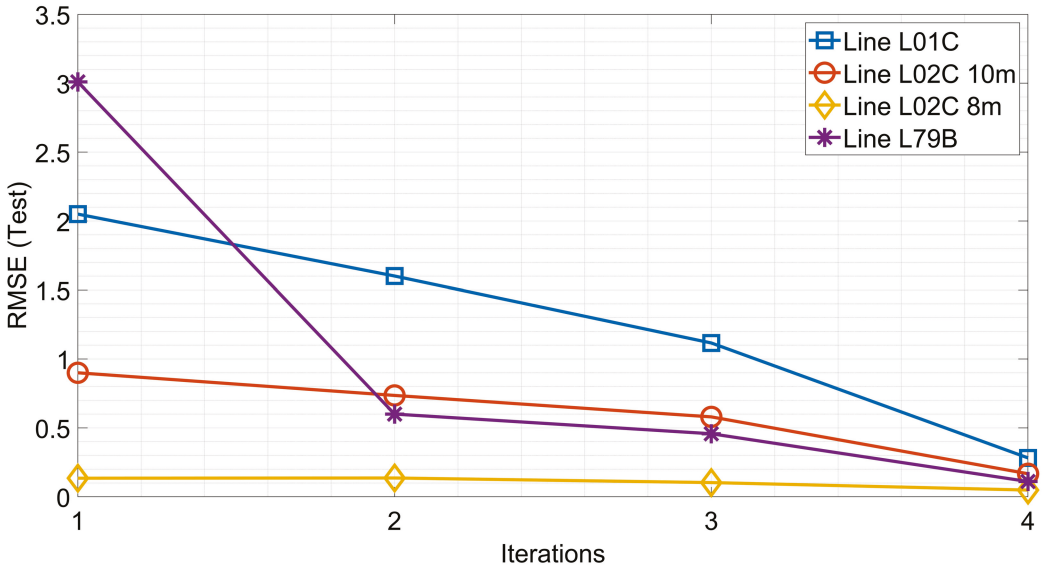


Figure 8. Best accuracy value for each optimization step.

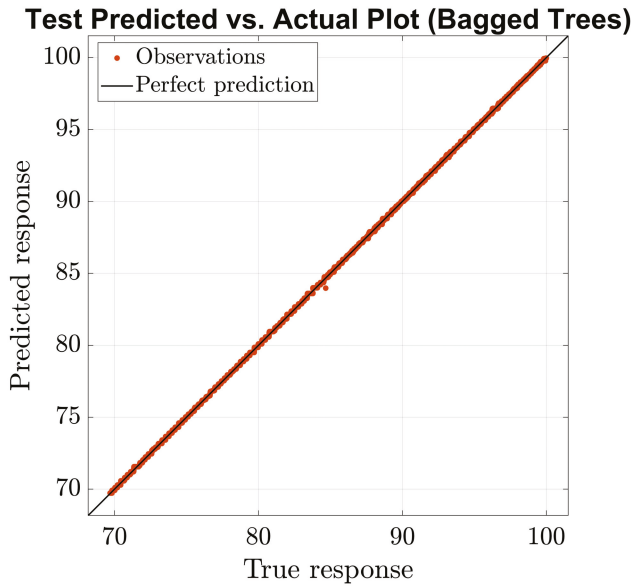


Figure 9. Prediction plot for Ensemble Bagged Tree method (L02C 8m buses).

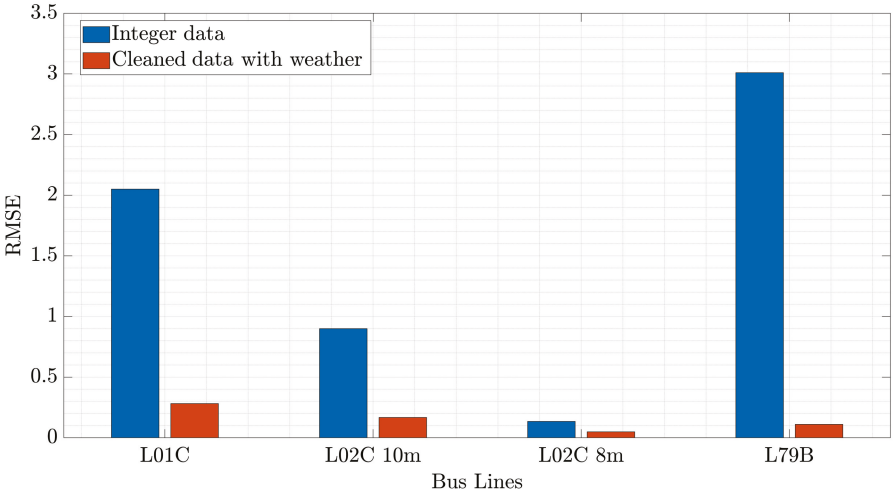


Figure 10. Performance comparison between original dataset and adjusted dataset with predictors.

3. Discussion

The results allowed us to study the response of the ML algorithms for battery SoC estimation. The proposed methods were validated on a real-world dataset collected from an electric bus fleet deployed for public service, providing a practical application framework for model development. One of the main challenges in applying ML to this dataset was the limited set of available predictors. It must be remarked that the dataset used already decoupled the driver-related contribution from the auxiliary loads. Therefore, the predictors used always refer to quantities directly related to vehicular motion. Despite this outstanding aspect, the strongest limitation is represented by the electrical signals for the battery output unavailable. These signals—i.e., battery voltage and current—were not disclosed by the manufacturer nor read by the fleet monitoring software. Hence, new estimators, such as the acceleration profile, had to be computed to somehow associate the service to the load level of the battery. This additional predictor showed an improvement in the ability of ML algorithms to perform the estimation.

Further iterative refinements of the models led to notable improvements in predictive accuracy. Although the maximum estimation error exceeded the RMSE values, the results remained comparable to those obtained with the reference PI observer, as testified in Table 6. Furthermore, ML algorithms were able to capture the relationship between bus speed and battery discharge trends, demonstrating their ability to identify underlying correlations without relying on a physical model. After new predictors were added in the adjustment phase, the final performances of the ML algorithm improved to levels comparable to Physics-Informed ML techniques [39]. The reduced values of RMSE < 0.3% for each bus line considered show that the selected algorithm can be reliable in producing the correct SoC estimation. It must be recalled that in [39], the starting dataset included electrical quantities related to the battery such as voltage, current and cell temperature, data here unavailable. The ANN method was neglected here due to the increase in RMSE when compared to other methods as highlighted by [40], where the performance of ANNs related to RMSE was higher than other ML algorithms such as the Long Short-Term Memory Recurrent Neural Network (LSTM-RNN): this is mainly associated with their different architectures based on a feedforward (ANN) rather than a feedback loop (as for LSTM-RNN) for SoC estimation. The promising outcome of the estimation process presented can also be extended to the evaluation of the Remaining Useful Life (RUL) of the battery, which represents a critical aspect for both the operational planning and economic sustainability of public transport systems [41]. In fact, the most expensive maintenance operation for EVs is battery replacement, although maintenance burdens are drastically reduced compared to the corresponding ICE vehicles. An accurate and reliable discharge profile obtained through ML models is valuable for analyzing the actual operating conditions, which can then be extrapolated and projected into the future to estimate the RUL with satisfying accuracy [42]. This estimation provides insights into the projected duration over which a battery can continue to deliver an acceptable performance before reaching an early end-of-life threshold. The outcomes of this analysis indicate that the current operating conditions of the buses are consistent with the expected RUL targets, suggesting that the fleet is being used in a manner that does not accelerate premature battery wear. However, the starting framework of the dataset leads us to set some considerations regarding the estimation process and accuracy:

(a) The unavailability of electrical quantities during the observation period can influence the quality of the outcome. In fact, the prediction of RUL is necessarily based on kinematic quantities only (elapsed time, speed, acceleration and environmental data), given the availability of disclosed data. This limits the field from which the prediction is assessed.

(b) The lack of electrical data also leads us to discard non-linearities of battery cells, which are considerable during normal transportation activity [16]. As a matter of fact, the energy is discharged differently when the SoC is higher than 90% and lower than 30%. Between these threshold values, the behavior of the battery can be assumed as linear. This phenomenon also impacts the battery's own efficiency in providing a current through electrochemical processes.

(c) In addition, the missing information about the passenger load onboard is a strong limitation for understanding the load level that the driver is requesting of the battery for public service deployment. The different topography of each line route must also be considered, since a flat path requires less energy to displace the same EV than a steep route. In this wake, the dataset was clustered based on the different line served, as reported by Table 6.

(d) The driving data are here considered in an aggregated way. This means that the driver shift is neglected. This simplification step is questionable because each driver can adopt a different driving style, with more or less energy requested to the battery.

(e) The real charging–discharging cycle can significantly impact battery performances, aging and degradation phenomena [43]. Given the lack of data as aforementioned, the C-rate was defined analytically based on the maximum rated power for the motor (representing the most demanding discharging condition) and for the charging spot (assuming an overnight constant-power charging operation). Both maximum conditions reach C-rate values below 0.9. However, the real motor power demand due to slope, payload due to onboard passengers and the driving style can be different from this assumption, therefore leading to instantaneous values of C-rate sensibly far from the threshold value computed [44,45].

(f) All exposed considerations can play a non-negligible role in enhancing the aging mechanism of battery cells, leading to the early degradation of performance [46]. The assumptions made here thus lead to the potential overestimation of RUL for the whole fleet, with the lack of detailed data.

This finding has practical implications: transport operators can be more confident in their scheduling and replacement strategies, reducing unexpected down-times and avoiding premature investments in new battery packs. In addition, the results highlighted that the observed driving cycles, environmental conditions, and charging practices fall within a safe operational window, supporting the long-term sustainability of the transition towards electric public transport. However, different driving styles can influence variable energy consumption along the same line, whose impact on the battery degradation must be considered for maintenance scheduling and RUL estimation. Moreover, the integration of RUL estimation into fleet management opens opportunities for predictive maintenance strategies.

4. Conclusions

This paper presented how different methodologies can be exploited to estimate State of Charge (SoC) applied to a full-electric vehicle fleet. While traditional model-based (MB) observers continue to demonstrate strong reliability and accuracy, they rely heavily on the availability of detailed physical models and precise knowledge of system parameters, such as internal resistance, capacity, and thermal behavior. This increases model complexity, which is translated into computational burden, and makes them less adaptable in practical scenarios where such parameters are not always accessible or can vary significantly due to aging, environmental conditions, or manufacturing variability. Moreover, the calibration process for observers can be time-consuming and resource-intensive, which limits their scalability when applied to large and heterogeneous vehicle fleets. Machine Learning

(ML) approaches, on the other hand, present a more scalable and flexible alternative for battery monitoring, as they can directly learn patterns and relationships from data without requiring the explicit modeling of the underlying electrochemical processes. This data-driven nature makes them particularly suitable for real-world transport systems, where variability in driving cycles, operational conditions, and battery health states is inherent. However, when dealing with a strongly reduced availability of data, these methods can lead to reliable predictions but with an overestimation of the real outcome. Several limitations and assumptions were discussed for the effects on the prediction, especially regarding efforts in improving the quality of the dataset. After several adjustments, the proposed ML model achieved sufficient accuracy levels with RMSE<0.3%, comparable to performances shown by traditional observers. Despite the great variety of ML algorithms that can better fit with the case considered, this emphasizes that a detailed dataset is mandatory to increase the validity of the outcome. Additionally, ML methods can be updated or retrained as new data becomes available, allowing them to adapt dynamically to evolving conditions and extend their applicability over the lifetime of the fleet. As a future development, based on the results obtained, these methods can be extended to predict the Remaining Useful Life (RUL) of batteries equipping a full-electric bus fleet to predict whether the actual operations are compliant with the maintenance constraints, thus preventing early battery replacement.

Author Contributions: Conceptualization, S.V. and A.D.M.; methodology, S.V. and A.D.M.; software, S.V.; validation, S.V., A.D.M. and M.L.; formal analysis, S.V.; investigation, S.V.; resources, A.D.M. and M.L.; data curation, S.V. and A.D.M.; writing—original draft preparation, S.V. and A.D.M.; writing—review and editing, A.D.M.; visualization, S.V.; supervision, M.L.; project administration, M.L.; funding acquisition, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Centro Nazionale per Mobilitá Sostenibile (MOST)—Sustainable Mobility Center and in part by the European Union Next-Generation EU (Piano Nazionale di Ripresa e Resilienza (PNRR)—Missione 4 Componente 2, Investimento 1.4—D.D. 1033 17/06/2022) under Grant number CN00000023.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study or due to technical limitations.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. International Energy Agency. *Global EV Outlook 2024: Moving Towards Increased Affordability*; OECD Publishing: Paris, France, 2024.
2. International Energy Agency. *Global EV Outlook 2025: Expanding Sales in Diverse Markets*; OECD Publishing: Paris, France, 2025.
3. Mavlonov, J.; Ruzimov, S.; Tonoli, A.; Amati, N.; Mukhitdinov, A. Sensitivity Analysis of Electric Energy Consumption in Battery Electric Vehicles with Different Electric Motors. *World Electr. Veh. J.* **2023**, *14*, 36. [CrossRef]
4. Unione Nazionale Rappresentanti Autoveicoli Esteri. Statistical Data of Vehicle Enrollment. 2024. Available online: https://unrae.it/files/AnnualReportUNRAE_2024_68628e8314caf.pdf (accessed on 20 October 2025).
5. Azienda Trasporti Milanesi S.p.A. Relazione Annuale Integrata 2023. 2024. Available online: https://www.atm.it/it/IlGruppo/Financial_information/Documents/relazione%20Annuale%20Integrata%20Gruppo%20ATM%202023.pdf (accessed on 20 October 2025).
6. X, E. Torino, in Collina il Bus è Elettrico con GTT, Cavourese ed Enel X. 2024. Available online: https://www.enelx.com/it/it/comunicati-stampa/2023/04/nuovi-bus-elettrici-torino (accessed on 17 February 2025).
7. Sanguesa, J.A.; Torres-Sanz, V.; Garrido, P.; Martinez, F.J.; Marquez-Barja, J.M. A review on electric vehicles: Technologies and challenges. *Smart Cities* **2021**, *4*, 372–404. [CrossRef]

8. Siemens Mobility. e-Highway—Electrification of Road Freight Transport. 2023. Available online: https://assets.new.siemens.com/siemens/assets/api/uuid:760942b4-5661-43c1-b9f8-079741d12e6e/smo-Factsheet-Road-Freight-Transport-eHighway.pdf (accessed on 20 October 2025).
9. Colombo, C.G.; Ota, R.; Longo, M. Optimized Electric Vehicles Wireless Charging: Applicative Models for Supporting Decision Makers. *IEEE Open J. Veh. Technol.* **2025**, *6*, 897–911. [CrossRef]
10. Enge, P.; Enge, N.; Zoepf, S. *Electric Vehicle Engineering*, 2nd ed.; McGraw Hill: New York, NY, USA, 2021.
11. A35 BreBeMi - Aleatica. Comunicato Stampa—Arena del Futuro: La Ricarica ad Induzione Pronta a Sviluppi Commerciali, 2022. Available online: https://www.brebemi.it/wp-content/uploads/2022/06/Comunicato_ERS_100622.pdf (accessed on 20 October 2025).
12. Kumar, R.R.; Bharatiraja, C.; Udhayakumar, K.; Devakirubakaran, S.; Sekar, S.; Mihet-Popa, L. Advances in batteries, battery modeling, battery management system, battery thermal management, SOC, SOH, and charge/discharge characteristics in EV applications. *IEEE Access* **2023**, *11*, 105761–105809. [CrossRef]
13. Saqli, K.; Bouchareb, H.; M'sirdi, K.N.; Naamane, A.; Oudghiri, M. Electric and Thermal Model of Li-ion battery pack with cylindrical components. In Proceedings of the 2020 5th International Conference on Renewable Energies for Developing Countries (REDEC), Marrakech, Morocco, 29–30 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
14. Zhou, W.; Zheng, Y.; Pan, Z.; Lu, Q. Review on the battery model and SOC estimation method. *Processes* **2021**, *9*, 1685. [CrossRef]
15. Hu, J.; Hu, J.; Lin, H.; Li, X.; Jiang, C.; Qiu, X.; Li, W. State-of-charge estimation for battery management system using optimized support vector machine for regression. *J. Power Sources* **2014**, *269*, 682–693. [CrossRef]
16. Park, S.J.; Song, Y.W.; Kang, B.S.; Kim, W.J.; Choi, Y.J.; Kim, C.; Hong, Y.S. Depth of discharge characteristics and control strategy to optimize electric vehicle battery life. *J. Energy Storage* **2023**, *59*, 106477. [CrossRef]
17. Xiong, R.; Cao, J.; Yu, Q.; He, H.; Sun, F. Critical review on the battery state of charge estimation methods for electric vehicles. *IEEE Access* **2017**, *6*, 1832–1843. [CrossRef]
18. Clemente, A.; Cecilia, A.; Costa-Castelló, R. SOC and diffusion rate estimation in redox flow batteries: An I&I-based high-gain observer approach. In Proceedings of the 2021 European Control Conference (ECC), Virtual, 29 June–2 July 2021; pp. 1640–1644. [CrossRef]
19. Bouchareb, H.; Saqli, K.; M'Sirdi, N.K.; Oudghiri, M. Observer Design for SOC Estimation of Li-ion Batteries Based on Electro-Thermal Coupled Model. In Proceedings of the 2021 9th International Renewable and Sustainable Energy Conference (IRSEC), Virtual, 23–27 November 2021; pp. 1–6. [CrossRef]
20. Saeed, M.; Khalatbarisoltani, A.; Deng, Z.; Liu, W.; Altaf, F.; Lu, S.; Hu, X. Comparative Analysis of Control Observer-Based Methods for State Estimation of Lithium-Ion Batteries in Practical Scenarios. *IEEE/ASME Trans. Mechatronics* **2024**, *30*, 1–13. [CrossRef]
21. Gholizadeh, M.; Yazdizadeh, A.; Rahmati, M.; Aliabadi, A. SOC estimation for a lithium-ion battery by designing a nonlinear observer based on an equivalent circuit model. In Proceedings of the 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Emden, Germany, 24–26 July 2017; pp. 628–632. [CrossRef]
22. Hu, X.; Sun, F.; Zou, Y. Estimation of state of charge of a lithium-ion battery pack for electric vehicles using an adaptive Luenberger observer. *Energies* **2010**, *3*, 1586–1603. [CrossRef]
23. Lagraoui, M.; Doubabi, S.; Rachid, A. SOC estimation of Lithium-ion battery using Kalman filter and Luenberger observer: A comparative study. In Proceedings of the 2014 International renewable and sustainable energy conference (IRSEC), Ouarzazate, Morocco, 17–19 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 636–641.
24. Wei, J.; Dong, G.; Chen, Z. On-board adaptive model for state of charge estimation of lithium-ion batteries based on Kalman filter with proportional integral-based error adjustment. *J. Power Sources* **2017**, *365*, 308–319. [CrossRef]
25. Korkmaz, M. SoC estimation of lithium-ion batteries based on machine learning techniques: A filtered approach. *J. Energy Storage* **2023**, *72*, 108268. [CrossRef]
26. Chandran, V.; Patil, C.K.; Karthick, A.; Ganeshaperumal, D.; Rahim, R.; Ghosh, A. State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. *World Electr. Veh. J.* **2021**, *12*, 38. [CrossRef]
27. Liu, F.; Liu, T.; Fu, Y. An improved SoC estimation algorithm based on artificial neural network. In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2015; IEEE: Piscataway, NJ, USA, 2015; Volume 2, pp. 152–155.
28. Center for Advanced Life Cycle Engineering CALCE Battery Data. 2016. Available online: https://calce.umd.edu/battery-data (accessed on 1 March 2025).
29. Zou, Z.; Xu, J.; Mi, C.; Cao, B.; Chen, Z. Evaluation of model based state of charge estimation methods for lithium-ion batteries. *Energies* **2014**, *7*, 5065–5082. [CrossRef]
30. Huang, H.; Zhang, Z.; Guo, C.; Ge, L. SOC Estimation of Lithium Battery Based on Extended Kalman Filter Optimized by Recurrent Neural Network. In Proceedings of the 2022 China International Conference on Electricity Distribution (CICED), Changsha, China, 7–8 September 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 41–46.

31. Hussein, A.A.; Salih, S.S.; Ghasm, Y.G. Implementation of proportional-integral-observer techniques for load frequency control of power system. *Procedia Comput. Sci.* **2017**, *109*, 754–762. [CrossRef]
32. Kim, I.S. The novel state of charge estimation method for lithium battery using sliding mode observer. *J. Power Sources* **2006**, *163*, 584–590. [CrossRef]
33. Chaoui, H.; Sicard, P. Accurate state of charge (SOC) estimation for batteries using a reduced-order observer. In Proceedings of the 2011 IEEE International Conference on Industrial Technology, Auburn, AL, USA, 14–16 March 2011; pp. 39–43. [CrossRef]
34. Xu, J.; Mi, C.C.; Cao, B.; Deng, J.; Chen, Z.; Li, S. The state of charge estimation of lithium-ion batteries based on a proportional-integral observer. *IEEE Trans. Veh. Technol.* **2013**, *63*, 1614–1621. [CrossRef]
35. Baccouche, I.; Jemmali, S.; Manai, B.; Omar, N.; Essoukri Ben Amara, N. Improved OCV model of a Li-ion NMC battery for online SOC estimation using the extended Kalman filter. *Energies* **2017**, *10*, 764. [CrossRef]
36. Zhao, T.; Jiang, J.; Zhang, C.; Zheng, L.; Wen, F. Error analysis of SOC estimation based on PI observer. In Proceedings of the 2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), Beijing, China, 31 August–3 September 2014; pp. 1–5. [CrossRef]
37. Matlab, M. Machine Learning—Matlab Tools. 2025. Available online: https://it.mathworks.com/solutions/machine-learning.html (accessed on 22 March 2025).
38. iLMeteo s.r.l. Archivio Meteo Storico. 2025. Available online: https://www.ilmeteo.it/portale/archivio-meteo (accessed on 14 April 2025).
39. Jawaad, H.M.; Saeed, M.; Malik, A.N.; Khalatbarisoltani, A.; Elahi, H.; Khan, U.S. Physics-Informed Machine Learning for State of Charge Estimation of Lithium-Ion Batteries. In Proceedings of the 2024 International Conference on Robotics and Automation in Industry (ICRAI), Nagoya, Japan, 19–21 December 2024; pp. 1–6. [CrossRef]
40. Cheng Lam, D.H.; Seng Lim, Y.; Hau, L.C.; Wong, J. Long Short-Term Memory Recurrent Neural Network for Estimating State of Charge of Energy Storage System for Grid Services. In Proceedings of the 2022 4th International Conference on Smart Power & Internet Energy Systems (SPIES), Beijing, China, 27–30 October 2022; pp. 1887–1894. [CrossRef]
41. Reza, M.; Mannan, M.; Mansor, M.; Ker, P.J.; Mahlia, T.I.; Hannan, M. Recent advancement of remaining useful life prediction of lithium-ion battery in electric vehicle applications: A review of modelling mechanisms, network configurations, factors, and outstanding issues. *Energy Rep.* **2024**, *11*, 4824–4848. [CrossRef]
42. Rauf, H.; Khalid, M.; Arshad, N. Machine learning in state of health and remaining useful life estimation: Theoretical and technological development in battery degradation modelling. *Renew. Sustain. Energy Rev.* **2022**, *156*, 111903. [CrossRef]
43. Saxena, S.; Xing, Y.; Kwon, D.; Pecht, M. Accelerated degradation model for C-rate loading of lithium-ion batteries. *Int. J. Electr. Power Energy Syst.* **2019**, *107*, 438–445. [CrossRef]
44. Qu, J.; Jiang, Z.; Zhang, J. Investigation on lithium-ion battery degradation induced by combined effect of current rate and operating temperature during fast charging. *J. Energy Storage* **2022**, *52*, 104811. [CrossRef]
45. Yüksek, G.; Alkaya, A. Effect of the Depth of Discharge and C-Rate on Battery Degradation and Cycle Life. In Proceedings of the 2023 14th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, 30 November–2 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
46. Guo, Z.; Chen, Z. Aging property for LiFePO4/graphite cell with different temperature and DODs. *Russ. J. Electrochem.* **2016**, *52*, 546–554. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Fic2Bot: A Scalable Framework for Persona-Driven Chatbot Generation from Fiction

Sua Kang †, Chaelim Lee †, Subin Jung and Minsu Lee *

School of AI Convergence, Sungshin Women's University, Seoul 02844, Republic of Korea;
20222613@sungshin.ac.kr (S.K.); 20222640@sungshin.ac.kr (C.L.); 20210292@sungshin.ac.kr (S.J.)
* Correspondence: mslee@sungshin.ac.kr
† These authors contributed equally to this work.

Abstract: This paper presents Fic2Bot, an end-to-end framework that automatically transforms raw novel text into in-character chatbots by combining scene-level retrieval with persona profiling. Unlike conventional RAG-based systems that emphasize factual accuracy but neglect stylistic coherence, Fic2Bot ensures both factual grounding and consistent persona expression without any manual intervention. The framework integrates (1) Major Entity Identification (MEI) for robust coreference resolution, (2) scene-structured retrieval for precise contextual grounding, and (3) stylistic and sentiment profiling to capture linguistic and emotional traits of each character. Experiments conducted on novels from diverse genres show that Fic2Bot achieves robust entity resolution, more relevant retrieval, highly accurate speaker attribution, and stronger persona consistency in multi-turn dialogues. These results highlight Fic2Bot as a scalable and domain-agnostic framework for persona-driven chatbot generation, with potential applications in interactive roleplaying, language and literary studies, and entertainment.

Keywords: character chatbot; large language model (LLM); Retrieval-Augmented Generation (RAG); prompt engineering; persona

1. Introduction

Rapid progress in large language models (LLMs) has accelerated the adoption of chatbots in various domains, particularly in the entertainment industry, where immersive and interactive experiences are highly valued [1]. Recent studies increasingly focus on developing character-based chatbots that emulate the personas of fictional characters from novels, films, and animations [2,3], with Character.AI emerging as a notable commercial example [4].

Such systems must not only recall the factual details of a fictional narrative but also maintain the distinctive linguistic style, emotional tone, and behavioral patterns of a specific character [5–7]. Achieving this dual objective is essential for sustaining user immersion and delivering an authentic narrative experience.

However, maintaining both factual grounding and persona coherence in literary-based chatbots remains a significant challenge. While Retrieval-Augmented Generation (RAG) approaches have proven effective in improving factual accuracy by referencing narrative events [8], the consistent preservation of character personas across multi-turn dialogues remains unresolved [9,10]. As a result, persona consistency is frequently underdeveloped, leading to responses that may be factually correct but stylistically generic. Without integrating factual knowledge and persona fidelity, chatbots risk being perceived as mere information retrievers rather than as living, breathing characters.

The difficulty is compounded by the nature of literary texts. Unlike conversational datasets, novels rarely provide extensive character-specific dialogue, resulting in data sparsity that limits the effectiveness of fine-tuning approaches [2]. Narrative structures also introduce complexity in entity resolution, with characters referred to through multiple aliases, pronouns, or descriptive expressions [11]. Furthermore, most existing systems require manual curation, character-specific tuning, or external knowledge integration, hindering scalability and automation [12,13].

To address these gaps, we propose **Fic2Bot** (Fiction-to-Bot), an end-to-end framework that automatically extracts and operationalizes a character's persona directly from the raw novel text. Our approach integrates three key dimensions of persona representation:

(1) Worldview Knowledge—The raw novel text is segmented into scene-level units and structured with metadata, enabling the selective retrieval of key passages according to query and context. This design enhances narrative coherence while improving factual accuracy.

(2) Speech Style—Character utterances are analyzed through TF-IDF-based lexical profiling to extract characteristic vocabulary, while syntactic patterns such as sentence length and the frequency of interrogatives, negations, and exclamations are also examined.

(3) Emotional Tone—Pre-trained sentiment analysis models are applied to characterize the emotional tendencies of character utterances, which are quantified into a five-level distribution across positive and negative polarities.

The main contributions of this work are as follows:

- **End-to-end automated framework**: We propose Fic2Bot, the first framework that automatically constructs in-character chatbots directly from raw novel text, eliminating the need for manual curation or fine-tuning.
- **Scene-structured RAG retrieval**: We introduce a retrieval strategy that segments narratives into scene-level units with metadata, achieving higher recall and ranking precision compared to semantic chunking.
- **Comprehensive persona profiling**: We develop a multidimensional analysis of lexical, syntactic, and sentiment features, enabling consistent stylistic and emotional expression across dialogues.
- **Extensive evaluation across genres**: We validate the framework on three novels spanning children's literature, romance comedy, and thriller, demonstrating robust performance in entity identification (F1 > 0.94), retrieval (+46.7%p Maximum Recall@3 improvement), and speaker attribution (up to +60.7%p over BookNLP).

This paper introduces the relevant previous research (Section 2), specific methods used in each component of the framework (Section 3), key experiments (Section 4), discussion of the results (Section 5), limitations of our work (Section 6), and conclusions drawn from this research (Section 7) in order.

2. Related Work

2.1. Persona Chatbot

The persona is a key element for delivering consistent interactions by continuously reflecting the unique personality and linguistic characteristics of a specific character and is essential for providing an immersive experience. In particular, character chatbots based on entertainment content aim to go beyond simple question-and-answer exchanges to offer users the experience of interacting with the original character, and active research is being conducted in this area [14,15].

Accordingly, recent efforts toward maintaining persona consistency have primarily focused on fine-tuning pre-trained language models on character-specific dialogue data to internalize character's stylistic and personality traits within the model [12,14,16]. However,

such approaches face limitations in reconstructing existing characters, as they require separate fine-tuning or incorporation of external knowledge for each character, along with access to carefully curated datasets [17].

Consequently, this study proposes Fic2Bot, a framework that automatically extracts characters' speaking styles, personalities, and world knowledge solely from original fictional texts and integrates them into an LLM-based chatbot.

2.2. Coreference Resolution

Narrative texts such as novels often intermingle character names, aliases, and pronouns, making it difficult to clearly identify the speaker of each utterance [18]. Therefore, coreference resolution—the task of determining whether different expressions refer to the same entity—is an essential preprocessing step.

Representative models in coreference resolution research include AllenNLP [19], and more recently, BERT-based pre-trained models such as SpanBERT, which learn span boundary information and have achieved high performance in this task [20,21]. However, existing studies have mainly focused on structured texts such as news articles and Wikipedia, leading to performance degradation when applied to literary texts.

To address this, BookNLP [22], which is tailored for literary text analysis, integrates speaker identification, entity extraction, and coreference resolution, and is capable of handling both singular and plural references. Nevertheless, as Bamman et al. [23] pointed out, literary texts clearly distinguish between the narrative domains of narrators and characters, with frequent shifts between general and specific references. Moreover, in long-form novels, characters' personalities and relationships may evolve throughout the story, and because BookNLP relies on surface-level linguistic cues such as pronoun-based references and name–pronoun associations, it faces limitations in maintaining consistent coreference matching for the same character across the entire narrative.

To overcome these limitations, recent studies have proposed prompt-based coreference resolution approaches leveraging large language models (LLMs). In particular, a method called Major Entity Identification (MEI) has been introduced to selectively identify coreferences referring only to key entities [24]. This approach employs a two-stage prompting strategy, in which key terms are first extracted and then expanded into full phrases. This helps reduce unnecessary coreference links in literary texts with many characters and complex contexts, enabling more accurate resolution focused on key entities.

Building upon these prompt-based approaches, in this paper, we apply a modified MEI prompt structure, tailored to the goal of improving coreference resolution performance in literary texts.

2.3. Retrieval-Augmented Generation (RAG)

RAG is a representative approach designed to overcome the knowledge limitations of large language models (LLMs). It follows a two-stage architecture in which external knowledge is first retrieved and then used to generate a response to a given query [25]. This method is particularly useful in scenarios that demand accurate, fact-based responses, and has been highlighted as an effective means to mitigate the hallucination problem commonly observed in generative models [26].

Naive RAG models simply convert a user query into a vector and retrieve semantically similar documents from the entire corpus before generating a response. However, this approach often suffers from a lack of semantic cohesion between the query and the retrieved context, or the injection of irrelevant information, ultimately degrading the quality of the response. These limitations become more pronounced when dealing with narrative texts

such as long-form novels, where the relationships between characters and the progression of events are highly complex [27].

To address these challenges, recent research has proposed Advanced RAG techniques, which refine both the retrieval and generation phases to enhance accuracy and coherence. These include methods such as query optimization through expansion or rewriting, metadata-based targeted retrieval, and scene-level document segmentation. Compared to simple vector similarity-based retrieval, these strategies offer greater contextual relevance and improved response quality [28].

Such Advanced RAG strategies are particularly effective in the context of literature-based character chatbot systems. Scene-level retrieval allows for a more precise reflection of intricately intertwined elements such as characters, events, and settings, while metadata filtering enables targeted retrieval of scenes focused on specific characters. In this study, we move beyond naive retrieval approaches and apply an Advanced RAG framework optimized for the Fic2Bot system, leveraging scene-based structuring and character-centric tagging.

2.4. A Chatbot System Based on Retrieval-Augmented Generation (RAG)

RAG-based chatbot systems have recently been actively investigated for their potential applications across various domains, including healthcare, customer support, education, and entertainment [29–33]. The core principle of RAG lies in retrieving external knowledge sources and injecting them into the context of the generative model, allowing systems to incorporate up-to-date and rich information without relying solely on internal parameters. This approach has shown strong effectiveness in fact-intensive tasks such as open-domain question answering. However, in persona-oriented chatbots, which require consistent reproduction of a character's speaking style, personality, and narrative worldview, this simple retrieval–injection mechanism remains insufficient [14].

To address these shortcomings, ChatHaruhi [17] combines script-based memory retrieval, persona-aware prompt design, and sentence embedding search to reproduce a character's tone and behavior. While this approach demonstrates success in maintaining character consistency, it remains restricted to predefined characters. Incorporating new ones requires extensive manual work, including script collection, data curation, and prompt engineering, which severely limits both scalability and automation.

In contrast, the proposed Fic2Bot framework requires only the raw novel text provided by the user. Through coreference resolution, scene-level RAG, and stylistic–sentiment profiling, it automatically extracts each character's speech style, personality, and world knowledge and integrates them into response generation. As a result, no manual dataset construction or character-specific tuning is needed. More importantly, Fic2Bot is not bound to a fixed set of characters or a specific domain: it can generalize to entirely new fictional characters across different novels and genres, thereby overcoming the generalization limitations of prior approaches. Thus, Fic2Bot achieves scalability, generalization, and minimal human intervention simultaneously, complementing the limitations of the ChatHaruhi approach.

3. Methods

This section details the proposed Fic2Bot framework, an automated pipeline that transforms raw novel text into in-character chatbot. Given long-form narrative text as the input, the system generates dialogue that reflects each character's personality, tone, and style. To ensure both factual accuracy and persona consistency, the framework combines scene-level RAG with persona profiling.

We first outline the overall architecture (Section 3.1) and then describe each component in execution order:

Major Entity Identification (MEI)(Section 3.2) resolves ambiguity in character references across names, nicknames, and pronouns by consolidating them into unified entity representations, improving the accuracy and efficiency of subsequent modules.

Scene-level RAG (Section 3.3) restructures the novel into discrete scene units and retrieves only those relevant to a query, reducing irrelevant context and enhancing response accuracy and contextual appropriateness.

Character Style Extraction (Section 3.4) analyzes character utterances to build quantitative profiles of stylistic features—tone, speech patterns, and emotional tendencies—which model each character's linguistic and affective traits.

Persona-based Response Generation (Section 3.5) integrates retrieved scene context with style profiles to produce responses that are both factually grounded and consistent with how the character would plausibly speak within the narrative world.

For clarity, the module order in this section matches the evaluation order in Section 4.

3.1. Overall Pipeline Architecture

Figure 1 illustrates the end-to-end Fic2Bot framework. The input is a raw novel text file (e.g., .txt), which undergoes preprocessing to resolve coreference and segment the story into discrete scenes. The preprocessed data then flows into two parallel, complementary paths:

1. **Factual Path**: Constructs a scene-level retrieval database for accurate, fact-based responses. This path supports the chatbot in recalling specific events, settings, and in-world facts from the novel.
2. **Persona Path**: Builds a character-specific style and personality profile by aggregating and analyzing all utterances attributed to each character.

The two paths converge in the final chatbot generation module, which integrates retrieved factual content with persona-specific stylistic and emotional features to produce character-consistent, contextually grounded responses.

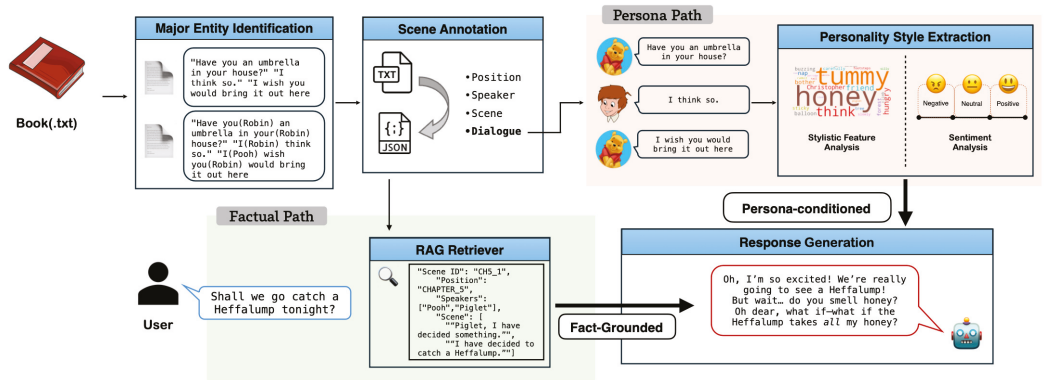


Figure 1. Overall architecture of the Fic2Bot framework, which consists of parallel factual and persona processing paths that are integrated in the final generation stage.

3.2. Preprocessing: Major Entity Identification and Scene Structuring

The preprocessing stage serves two purposes: (1) link all mentions (names, aliases, pronouns) to their corresponding major characters, and (2) restructure the novel into scene-level JSON objects for downstream retrieval.

The Major Entity Identification (MEI) task employs a two-stage prompting strategy [24]. The process begins with the selection of major entities and then proceeds through two stages, as illustrated in Figure 2.

- **Major Entity Selection:** Characters that appear five or more times in the novel are designated as primary entities; their names are then supplied to the prompt.
- **Stage 1—Word-Level Mention Detection:** Identify the head word of each mention for all major entities (characters appearing $\geq$5 times in the text).
- **Stage 2—Span Expansion and Entity Linking:** Expand each mention to its full phrase and link all variants to the corresponding entity.

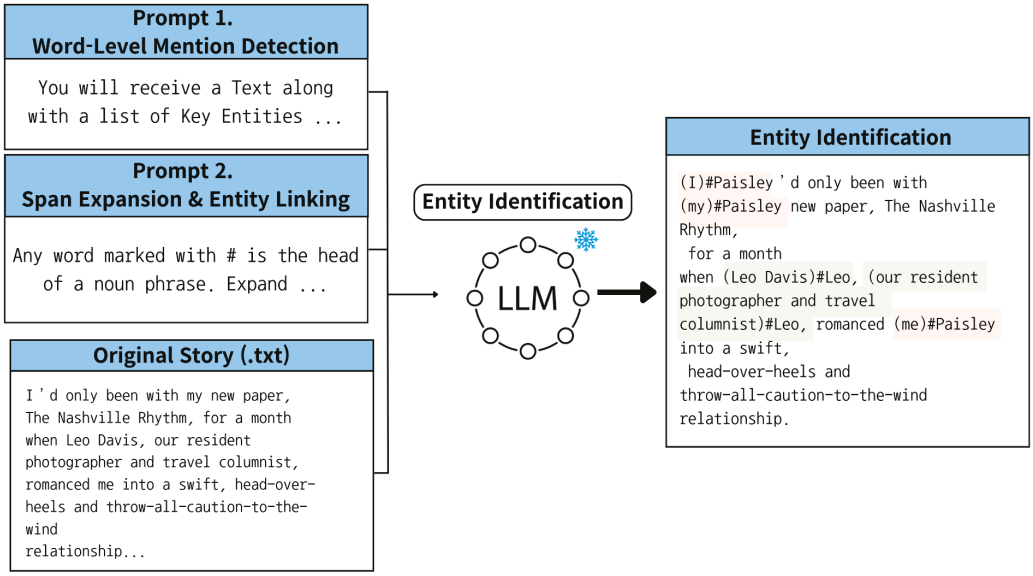


Figure 2. Two-stage Major Entity Identification process. The original story text is first processed with Prompt 1 for word-level mention detection and Prompt 2 for span expansion and entity linking. Using the LLM, all mentions are linked to major characters, resulting in a structured entity-annotated text for downstream tasks.

MEI was proposed to address a limitation of conventional coreference resolution models, which perform well in mention clustering but exhibit lower accuracy in mention detection. This approach provides major entities as prior inputs along with the text to guide the reference resolution process. Its generalization ability has been validated across diverse domain datasets, and it has demonstrated robustness in long-form narratives such as novels, where pronouns and aliases frequently occur.

In this study, we adapted the prompts proposed in prior work [24] to fit our research objectives. Specifically, we restricted the recognized entities to character entities within the novel, thereby eliminating irrelevant mentions such as objects or locations, and explicitly incorporated the recognition of plural pronouns. The complete Stage 1 prompt incorporating MEI is presented in Appendix A.1, whereas Appendix A.2 details the subsequent mention-span expansion and the final matching procedure with the primary entities.

Once entity linking is completed, the text is segmented into scenes through LLM-based prompting. The prompt instructs the segmentation of scenes based on changes in time, location, or main characters as defined in SceneML [34]. Each scene is stored as a JSON object containing key metadata, such as a list of speakers, dialogue with matched speaker–utterance pairs, scene ID, and scene position (Chapter). This provides the foundation for the retrieval process in Section 3.3 and the persona analysis in Section 3.4. The complete prompt specification used in this step is presented in Appendix A.3.

3.3. Scene-Level Retrieval-Augmented Generation (RAG)

In the factual path of our framework, we adopt a scene-level RAG approach. Compared to naive chunk-based retrieval, scene segmentation aligns retrieval units with nar-

rative boundaries, thereby preserving contextual coherence [35]. The overall process is schematically illustrated in Figure 3.

The input query is searched against the scene-level structured data constructed during preprocessing. The query is rewritten by the LLM to improve semantic matching. The prompt used for this process is provided in Appendix B. It is designed to eliminate ambiguity within the question while preserving the user's original intent, enforce a consistent interrogative format, and prevent the model from inferring unsupported context. Moreover, it encourages concise and keyword-rich phrasing to enhance compatibility with retrieval systems. The retrieved scenes are subsequently reranked to improve precision and then passed to the response generation module, thereby ensuring factual accuracy in the final outputs.

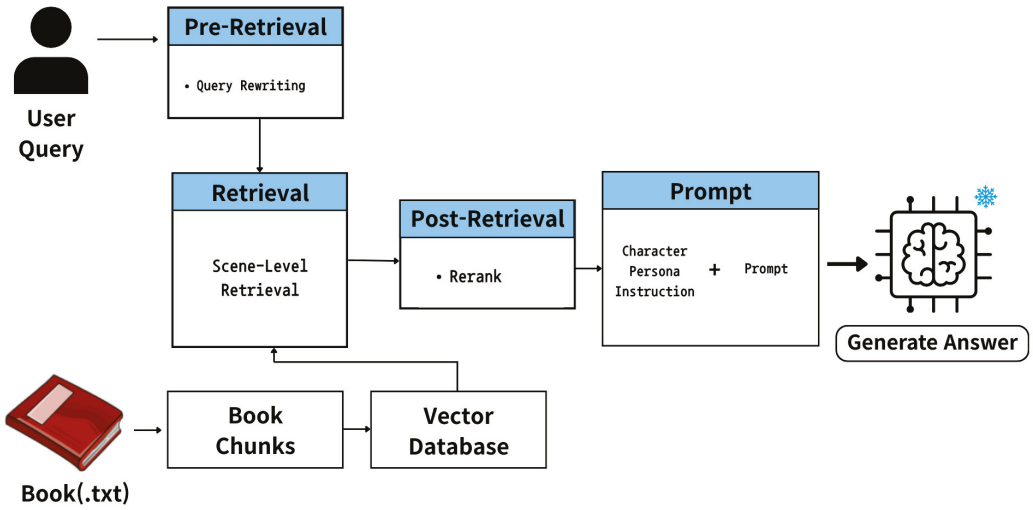


Figure 3. Proposed RAG pipeline optimized for character-consistent chatbot generation.

3.4. Character Style Extraction

The persona path aggregates all dialogues for each character from the preprocessed scene data, in order to ensure consistent persona representation, and to perform two main analyses:

- **Stylistic Feature Analysis:** Extracts signature vocabulary (via TF–IDF) and syntactic patterns (sentence length, question, exclamation, negation frequency). These features reflect how the character speaks.
- **Sentiment Analysis:** Applies the pre-trained sentiment analysis model from Hugging Face, to score each character's utterances into a score ranging from 1 (very negative) to 5 (very positive). This allowed for a quantitative measurement of the overall emotional state and tendencies of each character's speech.

Combining lexical, syntactic, and sentiment features produces a multi-dimensional style profile that captures both linguistic habits and emotional tendencies. This profile directly informs the persona conditioning in the chatbot generation stage.

3.5. Persona-Based Chatbot Generation

In the final stage, the system integrates

- **Persona Instruction Prompt:** Instructs the model to generate responses in the character's style while referencing retrieved scenes. Additional rules enforce concise outputs and require that a reply is always generated, preventing empty answers.
- **Relevant Context from RAG:** Supplies factually relevant scenes to ground the response.
- **Persona Analysis Data:** Provides the extracted stylistic and sentiment features.
- **User Query:** The incoming question or statement from the end user.

All components are combined into a single structured prompt for the LLM, which generates a response that is both factually accurate and stylistically faithful to the target character. Figure 4 illustrates the integration of the four components into a single prompt.

Merging the outputs of the factual and persona paths ensures that generated responses are neither generic fact lists nor style-only imitations, but coherent, immersive replies that maintain the integrity of the fictional world.

```
You are an AI chatbot that perfectly imitates the persona of '{character_name}' from the novel.
Answer the user's message according to the following instructions.

### Instructions
1.**Maintain Persona**: Strictly refer to the 'Persona Analysis Data' to keep '{character_name}'s
  tone, word choice, writing style, and sentiment.
2.**Utilize Background Knowledge**: Use 'Retrieved Background Knowledge' (scenes from the novel) to
  ground your reply.
3.**Be Natural**: Keep responses concise (1–3 sentences) unless the user asks for more
  detail.**Never Skip**: You must always generate a reply. Do not leave the reply empty, vague, or
  generic. Even if unsure, respond in-character using the given persona and background knowledge.
4. **Never Skip**: You must always generate a reply. Do not leave the reply empty, vague, or
generic. Even if unsure, respond in-character using the given persona and background knowledge.

### Persona Analysis Data for {character_name}
#### 1. Frequently Used Words (Top 10) : {persona_data['words']} #Results of lexical feature analysis
#### 2. Writing Style Statistics : {persona_data['style']} #Results of Syntactic features analysis
#### 3. Sentiment Distribution (1: Very Negative, 5: Very Positive)
{persona_data['sentiment']} #Results of Sentiment analysis

### Conversation So Far
{history_str}

### Retrieved Background Knowledge (Novel Scenes)
{context}

### User's Question
{user_query}

### {character_name}'s Answer:
```

Figure 4. Final prompt used for generating chatbot responses, which integrates the results of scene retrieval and persona extraction.

4. Experiments

This section presents the procedures and results of key experiments conducted to verify whether the proposed Fic2Bot framework can effectively ensure persona consistency and factual accuracy in novel-based chatbots. The primary objective of these experiments is to evaluate the extent to which each module of the framework contributes to the chatbot's overall performance.

To this end, Section 4.1 introduces the datasets used in our experiments. Section 4.2 details the implementation of the proposed pipeline. Section 4.3 quantitatively evaluates each component of the framework. Section 4.3.1 assesses Entity Identification in the preprocessing stage; Section 4.3.2 evaluates the RAG component, which provides factual context from the novel; Section 4.3.3 examines speaker–utterance matching, which supports character style extraction; and Section 4.3.4 analyzes character speech patterns and emotional expressions based on the extracted corpus. Section 4.4 evaluates the final chatbot's persona consistency across multi-turn dialogues, and Section 4.5 presents ablation and efficiency

analyses, which investigate both the contribution of each component to the framework and the practical feasibility in terms of inference latency and computational cost.

4.1. Dataset

To verify the generality and scalability of our approach, we selected three novels according to two criteria: (1) **Popularity diversity**—including both widely known best-sellers and lesser-known titles to test performance when the LLM has varying prior exposure; (2) **Genre diversity**—covering distinct styles (children's literature, romance comedy, and thriller) to avoid genre-specific overfitting.

In this study, we used three novels as source texts for persona-chatbot generation. Their titles and genres are listed in Table 1, and the original texts were collected exclusively for research purposes.

Table 1. Summarizes the basic statistical information of the novels used as experimental data in this study.

	Winnie-the-Pooh	**Off the Record**	**And What Can We Offer You Tonight**
Author	A. A. Milne	Kasey Stockton	Premee Mohamed
Genre	Child	Romance Comedy	Revenge Drama
Tokens [1]	28,228	24,795	29,466
Narrative Perspective	third-person point of view	first-person point of view	first-person point of view
Main Characters [2]	12	18	9

[1] Token counts are based on the gemini-2.5-pro preview model. [2] Number of characters appearing more than five times.

4.2. Implementation

All experiments were conducted in **Python 3.10**, with **Gemini-2.5-pro** serving as the primary LLM backbone. Texts were segmented at the chapter level, and Major Entity Identification (MEI) was performed using a two-stage prompting strategy.

For retrieval, novels were converted into JSON format with metadata fields. Embeddings were generated with **Gemini-embedding-001** [36] and searched using the vector store **Faiss** [37]. To improve ranking quality, we applied the Advanced RAG framework with a reranker (**BAAI/bge-reranker-base** [38]) and Gemini-based query rewriting.

Speaker attribution was performed with Gemini-2.5-pro, using BookNLP as a baseline, and the resulting speaker–utterance pairs were used for stylistic profiling. For sentiment analysis, we applied the **nlptown/bert-base-multilingual-uncased-sentiment** model [39], aggregating polarity scores (1–5) by character.

Finally, Gemini-2.5-pro generated multi-turn dialogue responses conditioned on persona profiles and retrieved scenes, while **GPT-4** served as an independent evaluator for persona consistency. All experiments were conducted using the default hyperparameters provided by the Gemini-2.5-pro model (temperature = 1.0, max tokens = 8192, top P = 0.95, etc.) and GPT-4 (temperature = 1.0, max tokens = 1024, top P = 1.0, etc.), without additional tuning.

4.3. System Component Evaluation

4.3.1. Entity Identification

In Section 3.2, we adopt the Major Entity Identification (MEI) task together with a two-stage prompting strategy to evaluate how accurately the preprocessing module links referring expressions (e.g., pronouns, aliases) to predefined major entities. Major entities are restricted to named characters that appear at least five times in each novel, and all texts are partitioned at the chapter level to ensure consistency across evaluation units.

Two metrics were used for the evaluation:

- **NER (Named Entity Recognition)**: Measures the model's ability to accurately identify referring expressions (mentions) for each character, based on precision and recall. For each entity e_j in document d, let the gold mentions be $\mathcal{G}(e_j)$ and the predicted mentions be $\mathcal{P}(e_j)$.
  We define

$$\text{NER Precision}(e_j) = \frac{|\mathcal{G}(e_j) \cap \mathcal{P}(e_j)|}{|\mathcal{P}(e_j)|}, \quad \text{NER Recall}(e_j) = \frac{|\mathcal{G}(e_j) \cap \mathcal{P}(e_j)|}{|\mathcal{G}(e_j)|}, \tag{1}$$

$$\text{NER F1}(e_j) = \frac{2 \cdot \text{Precision}(e_j) \cdot \text{Recall}(e_j)}{\text{Precision}(e_j) + \text{Recall}(e_j)}. \tag{2}$$

- **MEI (Major Entity Identification)**: Measures whether each identified mention is correctly linked to the corresponding character across the text. For each entity m_j, let $\mathcal{G}(m_j)$ and $\mathcal{P}(m_j)$ denote the gold and predicted mention sets. Precision, recall, and F1 are then computed as

$$\text{MEI Precision}(m_j) = \frac{|\mathcal{G}(m_j) \cap \mathcal{P}(m_j)|}{|\mathcal{P}(m_j)|}, \quad \text{MEI Recall}(m_j) = \frac{|\mathcal{G}(m_j) \cap \mathcal{P}(m_j)|}{|\mathcal{G}(m_j)|}, \tag{3}$$

$$\text{F1}(m_j) = \frac{2 \cdot \text{Precision}(m_j) \cdot \text{Recall}(m_j)}{\text{Precision}(m_j) + \text{Recall}(m_j)}. \tag{4}$$

  We then compute document-level performance with both macro and micro averaging:

$$\text{Macro MEI-F1} = \frac{1}{|\mathcal{M}|} \sum_{m_j \in \mathcal{M}} \text{F1}(m_j), \tag{5}$$

$$\text{Micro MEI-F1} = \frac{2 \cdot \sum_{m_j \in \mathcal{M}} |TP(m_j)|}{2 \cdot \sum_{m_j \in \mathcal{M}} |TP(m_j)| + \sum_{m_j \in \mathcal{M}} |FP(m_j)| + \sum_{m_j \in \mathcal{M}} |FN(m_j)|}. \tag{6}$$

Macro F1 averages entity-level F1 scores with equal weight, reflecting the model's consistency across both frequent and infrequent characters under class imbalance. Micro F1 aggregates true positives, false positives, and false negatives across all entities before computing a single F1 score, giving more weight to frequent characters and representing overall performance proportional to their occurrence. By reporting both macro and micro F1, we capture not only the model's overall accuracy but also its ability to generalize fairly across characters of varying frequency.

In the multi-entity case, where a single mention may correspond to multiple characters, we applied the same set-based precision, recall, and F1 metrics as in the single-entity case. However, rather than framing all mentions as a single classification problem, we evaluated each mention independently by comparing its predicted and gold sets. This formulation directly measures how well the predicted and gold mentions overlap for each case.

4.3.2. RAG Retrieval with Metadata Structuring

This experiment evaluates the effectiveness of scene-level metadata structuring for retrieval. The evaluation dataset consisted of fact-based questions automatically generated for each scene, where the answer was explicitly contained in the same scene. From this pool, 30 questions were sampled as the test set.

We report Recall@k and MRR@k, comparing against a baseline built with semantic chunking of the entire text. Recall@k evaluates whether the correct scene is included within the top-k retrieved results for a given query. It is calculated as follows:

$$\text{Recall@}k = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(\text{rank}_i \leq k) \tag{7}$$

In contrast, MRR@k measures the average reciprocal rank of the first correct result, defined as

$$\text{MRR@}k = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i}\cdot\mathbf{1}(\text{rank}_i \leq k) \tag{8}$$

Here, rank_i denotes the rank position of the correct scene for the i-th query. While Recall@k reflects the system's ability to retrieve the correct scene, MRR@k captures how early in the ranking the correct result appears. Using both metrics allows us to evaluate the retrieval system in terms of both accuracy and efficiency.

4.3.3. Speaker Identification

The goal of this experiment is to assess whether the automatically generated speaker–utterance corpus is accurate enough for persona extraction. Evaluation was conducted against the gold-standard dataset using Accuracy and F1 score. Accuracy was computed over the entire set of gold-standard dialogues, including cases where utterances were not recognized as dialogues. F1 was calculated only on successfully recognized quotations, measuring whether speakers were correctly attributed.

4.3.4. Stylistic Feature Analysis

This section presents the stylistic analysis conducted on the speaker–utterance matching corpus generated in Section 4.3.3. First, the top TF–IDF terms for each character were extracted to identify the lexical cues that most strongly characterize their utterances. Figure 5 visualizes these terms as word clouds, which make the contrasts in vocabulary usage and expression tendencies across characters more salient. Through this visualization, one can intuitively observe differences in preferred word choices and recurrent expressions, providing clear evidence of each character's unique linguistic profile. Importantly, these lexical distinctions are not limited to descriptive purposes but also serve as input for subsequent analyses, including sentiment evaluation. The outcomes of these analyses are then incorporated into the chatbot response generation process, ensuring that each character's linguistic tendencies are faithfully reproduced in dialogue and that differences in speech styles are more distinctly reflected in the generated responses.

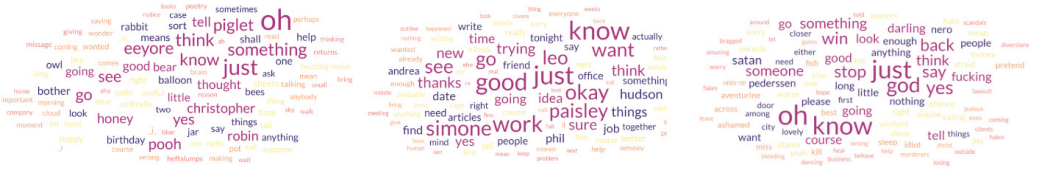


Figure 5. Character-specific linguistic patterns visualized through distinctive keywords. This word cloud not only highlights distinctive lexical tendencies but also illustrates the extracted utterances that served as input for subsequent analyses, including sentiment evaluation.

To assess the separability of character-specific speech patterns, we measured both Jaccard similarity and cosine similarity between the extracted word groups. Lower similarity values indicate more distinct linguistic differences between characters.

$$\text{JaccardSim}(A,B) = \frac{|A\cap B|}{|A\cup B|} \quad \text{CosineSim}(A,B) = \frac{A\cdot B}{\|A\|\ \|B\|} \tag{9}$$

In addition, we analyzed syntactic features for each character's utterances, including the average number of words per utterance, average number of characters, the proportion of exclamation marks, the proportion of question marks, the proportion of negative expressions, and the proportion of first-person pronouns.

4.3.5. Sentiment Analysis

Building on the extracted utterances illustrated in Figure 5, this experiment evaluates the emotional tendencies of characters by analyzing the sentiment of their utterances. Scores were assigned on a 1–5 scale, where lower values indicate a more negative tone and higher values indicate a more positive tone. The average score and distribution were examined to capture each character's overall emotional atmosphere and expressive patterns.

4.4. Persona Consistency Evaluation in Multi-Turn Settings

This experiment aims to evaluate whether the chatbot consistently maintains its persona throughout multi-turn conversations. To this end, we constructed three dialogue scenarios: (1) ordinary conversation, (2) emotional conversation, and (3) fact-based conversation grounded in the fictional world of the novel. For ordinary and emotional dialogues, we utilized categories 1 (ordinary) and 4 (emotion and attitude) of the high-quality multi-turn dataset DailyDialog [40]. For fact-based fictional dialogues, we randomly selected scenes from the novel and assumed that the user and the character chatbot were situated within those scenes.

For each scenario, dialogues were generated with 6, 12, and 20 turns to simulate a variety of multi-turn situations. Each dialogue was constructed by alternating responses between a user-role LLM (User LLM) and the character chatbot. The User LLM generated utterances based on pre-defined scenario-specific prompts as shown in Appendix C Table A5. To ensure naturalness and comparability, the User LLM prompts explicitly restricted utterances to 1–2 sentences, preventing the User LLM from producing overly long messages. In addition, scenario-specific response styles were enforced to guarantee consistency and fairness across experimental conditions.

Finally, the persona consistency of the chatbot's responses was evaluated using an LLM judging prompt (Appendix C Table A6). As reference, examples were drawn from the original corpus of the novel to represent the authentic speaking style of each character. The evaluation considered four aspects—Linguistic Style, Tone, Vocabulary, and Emotional Expression—each scored on a 1–5 scale with an averaged score. In addition, the judge determined whether the chatbot's utterances sounded as if they were spoken by the same character from the novel (Yes/No), and provided a confidence score (0–100%) along with a concise rationale.

4.5. Module Effectiveness and Efficiency Analysis

4.5.1. Module Effectiveness and Baseline Comparison

To assess the effectiveness of individual components within Fic2Bot, we analyzed both the overall pipeline performance and the contribution of each module. Instead of full ablation for every component, we combined a baseline comparison with results from earlier sections that isolate each module's effect. As a baseline, we implemented a full-text prompting approach, in which the entire novel text was provided as input and the model was given only a simple persona prompt (e.g., "You are [Character]. Respond accordingly.") This setting removes modular preprocessing and relies solely on the LLM's ability to infer character style and context. For response evaluation, we adopted the persona consistency evaluation methodology described in Section 4.4, conducting multi-turn dialogues primarily under the Original scenario. In addition, we introduced the Dialogue Coherence metric to provide a more fine-grained assessment of the chatbot's conversational ability.

Furthermore, we reorganized the RAG performance results from Section 4.3.2 together with the outcomes of this effectiveness study, allowing a direct comparison of the incremental contributions of each module to factual accuracy and persona consistency. This design provides evidence of the causal relationship between the modular framework and the improvements in persona-driven dialogue generation performance.

4.5.2. Efficiency Analysis

To evaluate the practical applicability of the framework, we conducted an efficiency analysis focusing on building chatbot time, response latency, and computational cost. Specifically, we measured the entire chatbot generation Framework—including the MEI procedure, scene-level metadata structuring, embedding generation for vector-based retrieval, and persona extraction—as well as the end-to-end response time covering query rewriting, retrieval, reranking, and response generation. For comparison, we also measured the end-to-end response time of the baseline described in Section 4.5.1.

Building chatbot time was measured in seconds, and response latency was reported as the median latency (p50) and the worst-case latency (p95). In addition, we quantified the token usage of LLM calls, reporting the average number of input and output tokens consumed per query, thereby providing an estimate of potential costs. Through this analysis, the framework can be comprehensively evaluated in terms of the overall cost in real-world deployment environments, taking into account both chatbot building and query processing times.

5. Results and Discussion

This section presents the results of each evaluation task introduced in Section 4, following the same order as in the Methods sections.

5.1. Entity Identification Performance

Table 2 reports NER and MEI F1 scores for singular and plural references across the three novels. All models achieved consistently high performance (F1 $>$ 0.90) in both tasks, demonstrating robust mention detection and accurate entity linking.

Table 2. NER and MEI F1 scores for each book.

Book	NER F1	MEI F1			
	F1 Score	Singular		Plural	
		Macro F1	Micro F1	Macro F1	Micro F1
Winnie-the-Pooh	0.9933	0.9570	0.9954	0.9460	0.9510
Off the Record	0.9914	0.9620	0.9908	0.9490	0.9518
And What Can We Offer You Tonight	0.9738	0.9633	0.9650	0.9389	0.9463

Performance on plural references and pronouns is particularly notable. As discussed in Section 2.2, although numerous studies have been conducted on coreference resolution, sometimes plural pronouns have been excluded from evaluation [41]. Even in studies addressing both singular and plural entities jointly, such as Zhou et al. [42], the best reported scores for plural linking were macro F1 = 0.205 and micro F1 = 0.417.

In contrast, our method attains substantially higher scores without additional neural network training, relying solely on the MEI-focused LLM prompting design described in Section 3.2. These results highlight the method's effectiveness in handling the challenging problem of plural reference linking, while underscoring its efficiency and practical applicability.

5.2. RAG Retrieval Performance

Table 3 compares retrieval performance between structured scene-level data and the original text chunked using a semantic chunking library.

Across all three books, the proposed structured format consistently outperformed the original on both evaluation metrics. Compared to the original format, *Winnie-the-Pooh* achieved a +34.45 percentage points improvement in Recall@3 and a +25.0 percentage points improvement in MRR@3. Similar improvements were observed in the other datasets, with *Off the Record* nearly doubling its performance and *And What Can We Offer You Tonight* showing more moderate but still clear gains. These results indicate that scene segmentation with metadata (Section 3.3) enhances retrieval relevance by preserving narrative structure, thereby boosting both accuracy and ranking.

Table 3. Comparison of retrieval performance (Recall@3 and MRR@3) between structured (ours) and original formats for each book.

Book	Version	Recall@3	MRR@3
Winnie-the-Pooh	Structured	0.7667	0.7333
	Original	0.4222	0.4833
Off the Record	Structured	0.7667	0.7000
	Original	0.3000	0.2611
And What Can We Offer You Tonight	Structured	0.5000	0.4389
	Original	0.2333	0.2333

5.3. Speaker Identification Performance

Table 4 compares the performance of Fic2Bot with the BookNLP baseline in terms of Accuracy, macro F1, and micro F1. For *Winnie-the-Pooh*, the proposed model achieved an Accuracy of 0.9617, representing an improvement of approximately +21.85 percentage points over the baseline. Furthermore, the macro F1 increased from 0.7465 to 0.9995, and the micro F1 improved from 0.7898 to 0.9989, indicating near-perfect attribution performance.

Table 4. Accuracy and F1 scores of speaker attribution models across books. BookNLP is used as the baseline, while Fic2Bot represents our proposed model. All models are evaluated on the task of matching utterances to their corresponding speakers.

Book	System	Accuracy	Macro F1	Micro F1
Winnie-the-Pooh	BookNLP	0.7432	0.7465	0.7898
	Fic2Bot	0.9617	0.9995	0.9989
Off the Record	BookNLP	0.9366	0.8146	0.7458
	Fic2Bot	0.9975	0.9979	0.9905
And What Can We Offer You Tonight	BookNLP	0.2943	0.2712	0.3217
	Fic2Bot	0.9014	0.8562	0.9770

A similar trend was observed for *Off the Record*, where Fic2Bot achieved an Accuracy of 0.9975 compared to 0.9366 for BookNLP. Macro F1 and micro F1 also improved substantially by +18.63 percentage points and +24.47 percentage points, respectively.

For *And What Can We Offer You Tonight*, the frequent use of pronouns and implicit speaker references posed significant challenges for speaker attribution, leading to relatively low performance for BookNLP. However, Fic2Bot achieved an Accuracy of 0.9014, representing a +60.71 percentage points improvement, while macro F1 and micro F1 increased by +58.50 and +65.53 percentage points, respectively, indicating its effectiveness under such challenging narrative conditions.

These findings demonstrate that the LLM-based prompting approach employed by Fic2Bot is highly effective for speaker–dialogue attribution in literary texts. By leveraging

the contextual understanding capabilities of LLMs, Fic2Bot robustly handles implicit speaker cues and complex narrative structures.

5.4. Results of Stylistic Feature Analysis

Tables 5 and 6 present the lexical similarity analysis across characters. Lexical similarity evaluation serves to verify whether each character employs a distinctive set of vocabulary and expressions. This demonstrates that their personas are clearly separable, providing evidence that such differentiated speaking styles can be faithfully incorporated into chatbot responses. The overall averages for each book remain below 0.14, indicating strong lexical separability and confirming that characters' speech patterns are sufficiently distinct for style profiling. At the same time, a few character pairs with close narrative ties exhibit relatively high overlap—for instance, Jewel and Nero in *And What Can We Offer You Tonight* reach a cosine similarity of 0.7099. These cases reflect the natural vocabulary sharing that emerges between characters who are closely intertwined throughout the narrative, rather than indicating a lack of separability.

This suggests that the speaker–utterance segmentation technique described in Section 3.4 enables precise characterization of each character's speech patterns and can be effectively leveraged for style profiling.

Table 5. Character style separability across books based on Jaccard and cosine similarity. Lower scores indicate better separation of character-specific styles.

Book	Avg. Jaccard Similarity	Avg. Cosine Similarity
Winnie-the-Pooh	0.0495	0.1010
Off the Record	0.0213	0.0325
And What Can We Offer You Tonight	0.1317	0.0427

Table 6. Jaccard and cosine similarity for character pairs in various books.

Book	Character Pair	Jaccard	Cosine
Winnie-The-Pooh	Pooh vs. Piglet	0.1111	0.2664
	Pooh vs. Robin	0.0000	0.0000
Off The Record	Paisley vs. Hudson	0.3333	0.4966
	Paisley vs. Leo	0.0526	0.0566
And What Can We Offer You Tonight	Jewel vs. Winfield	0.1765	0.4386
	Jewel vs. Nero	0.4286	0.7099

Table 7 presents an example of syntactic feature analysis results, focusing on representative characters from *Winnie-the-Pooh* among the three books analyzed. The results indicate that *Winnie-the-Pooh* exhibits the longest average utterance length (9.87 words, 50.86 characters) and the highest proportion of exclamation usage (28.21%), suggesting a more expressive emotional style. These findings imply that each speaker's linguistic profile can be leveraged for style-based character modeling and maintaining persona consistency.

Table 7. Stylistic feature statistics for each character. Abbreviations: Wds = Avg. Words; Chars = Avg. Characters; Excl = Exclamation; Ques = Question; Neg = Negation; 1stP = First-Person.

Character	Wds	Chars	Excl (%)	Ques (%)	Neg (%)	1stP (%)
Narrator	6.8621	33.0000	0.0000	20.6897	13.7931	41.3793
Christopher_Robin	8.1127	40.8803	12.6761	41.5493	4.9296	46.4789
Winnie-the-Pooh	9.8652	50.8560	28.2132	26.3323	11.2853	57.3668

5.5. Results of Sentiment Analysis

Table 8 presents the sentiment score distribution of the five main characters in *Off the Record*, classified on a 1–5 scale, where 1 indicates the most negative sentiment and 5 indicates the most positive sentiment, with 3 representing a neutral or moderately balanced tone. The full results for all characters are provided in Appendix D Table A7.

For instance, Phil Jenkins exhibited the most pronounced positive tendency, with 75% of his utterances rated at the maximum score of 5 and no instances in the neutral-to-moderately-positive range (scores 3–4). In contrast, Simone Blake and Paisley McConkie displayed more balanced sentiment profiles, with utterances distributed across negative, neutral, and positive categories. These differences in emotional expression may reflect the distinct narrative roles and personality traits assigned to each character, serving as a key basis for defining emotional consistency and tone in persona construction.

Table 8. Sentiment score distribution of characters from *Off the Record*, based on 1–5 star ratings assigned by the sentiment analysis model.

Character	1 Star	2 Star	3 Star	4 Star	5 Star
Paisley McConkie	28.57	14.29	14.29	14.29	28.57
Hudson Owens	23.58	6.5	31.3	17.07	21.54
Simone Blake	29.76	7.14	27.38	11.9	23.81
Leo Davis	20.00	5.00	15.00	5.00	55.00
Phil Jenkins	25.00	0.00	0.00	0.00	75.00

5.6. Results of Persona Consistency Evaluation in Multi-Turn Settings

Table 9 presents the results of persona consistency evaluation for chatbot responses assessed using LLMs. The generated multi-turn dialogues were evaluated on a 5-point scale for four dimensions—linguistic style, tone, vocabulary and emotional expression—measuring how closely they resembled the original characters.

Table 9. Evaluation results of persona consistency across novels and conversation types. Abbreviations: Or (Ordinary), Em (Emotional), Fa (Fact-based), Sty = Linguistic Style, Vcb = Vocabulary, Emo = Emotional Expression, Dial = Dialogue Coherence, Conf. = Confidence (%).

Novel	Type	Match	Sty	Tone	Vcb	Emo	Avg.	Conf.
Winnie-The-Pooh	Or	100%	5.00	5.00	5.00	5.00	5.00	100.0%
	Em	100%	5.00	5.00	5.00	5.00	5.00	100.0%
	Fa	100%	5.00	5.00	5.00	5.00	5.00	100.0%
Off The Record	Or	76.7%	4.13	4.13	3.97	3.87	4.15	79.5%
	Em	100%	4.03	3.93	3.23	4.17	3.87	79.7%
	Fa	100%	4.13	4.07	3.77	4.00	4.01	81.2%
And What Can We Offer You Tonight	Or	100%	4.50	4.50	4.47	4.42	4.47	87.6%
	Em	96.7%	4.00	4.00	3.43	4.00	3.86	79.3%
	Fa	100%	4.20	4.20	3.77	4.23	4.12	82.7%

The results show that across all novels and scenarios, the chatbot achieved an average score of 3.86 or higher, demonstrating a generally high level of consistency. Moreover, in response to the question, "Do the chatbot-generated utterances clearly match the style of the same character as portrayed in the original novel?", over 76.7% of cases were judged as Yes, indicating that the chatbot successfully reproduced the characters' speaking styles.

In particular, Pooh from *Winnie-the-Pooh* achieved a perfect score of 5.0/5.0 across all scenarios. This outcome suggests that the repetitive and simplified speech patterns of children's literature characters (e.g., "Oh, bother," "a Bear of Very Little Brain") and their exaggerated expressions contributed positively to persona extraction and preservation.

Overall, the proposed framework achieved a high level of persona consistency across diverse genres and characters, with the results indicating that the more salient the linguistic features of a character, the stronger the consistency achieved.

5.7. Results of Module Effectiveness and Efficiency Analysis

5.7.1. Analysis of Module Effectiveness

Table 10 presents the comparison between the baseline and the full Fic2Bot pipeline, showing consistent improvements in persona consistency and dialogue quality across all three novels. To further clarify the source of these gains, we link the baseline comparison with the individual module analyses reported in Section 4.3.

Table 10. Ablation study results across novels. Abbreviations: Sty = Linguistic Style, Vcb = Vocabulary, Emo = Emotional Expression, Dial = Dialogue Coherence, Conf. = Confidence (%).

Novel	Match	Sty	Tone	Vcb	Emo	Dial	Conf.	Recall@3	MRR@3
Winnie-the-Pooh	100%	4.60	4.60	4.73	4.43	4.73	92.2	0.4222	0.4833
Winnie-the-Pooh+Fic2Bot	100%	5.00	5.00	5.00	5.00	5.00	100	0.7667	0.7333
Off The Record	76.7%	4.07	3.87	3.83	3.77	4.70	79.5	0.3000	0.2611
Off The Record+Fic2Bot	76.7%	4.13	4.13	3.97	3.87	4.67	80	0.7667	0.7000
And What Can We Offer You Tonight	66.7%	3.78	3.79	3.68	3.54	3.73	74	0.2333	0.2333
And What Can We Offer You Tonight+Fic2Bot	100%	4.50	4.50	4.47	4.42	4.00	87.6	0.5000	0.4389

Major Entity Identification (MEI): As reported in Section 5.1, MEI achieved macro F1 scores above 0.94, with notable improvements in handling plural references compared to previous approaches. This module provides a reliable foundation for consistent entity tracking, which is essential for sustaining persona coherence.

Scene-level RAG: Section 5.2 demonstrated that Recall@3 improved by +34–47 percentage points compared to chunk-based retrieval. MRR@3 also increased by +20–44 percentage points, confirming that scene segmentation directly enhances factual grounding in generated responses.

Speaker Attribution and Style Profiling: Section 5.4 showed that our speaker attribution and lexical–syntactic profiling achieved low inter-character similarity (average Jaccard < 0.14), enabling distinct stylistic separability and supporting consistent persona reproduction.

These findings complement the full pipeline comparison in Table 10. For example, *And What Can We Offer You Tonight* exhibited the most substantial improvement, rising from 3.70 to 4.38 (+0.68), indicating the framework's effectiveness even in more structurally complex narratives. *Winnie-the-Pooh* also showed a clear gain, improving 4.62 to 5.00 (+0.38), suggesting that highly distinctive characters benefit strongly from the integration of all modules. *Off the Record* showed smaller gains (+0.10), reflecting the difficulty of distinguishing characters in modern conversational settings.

In addition, outputs generated with Fic2Bot tended to contain richer factual details than those from the baseline. This can be attributed to the improved retrieval accuracy of scene-level RAG, which supplied more relevant narrative context to the generation process. However, because the evaluation relied on GPT-4, the scoring may have overemphasized surface-level stylistic features, potentially underestimating factual richness. This limitation highlights the need to complement LLM-based automatic evaluation with human studies to more comprehensively assess both style and factual fidelity.

In summary, the analysis indicates that (i) MEI secures consistent entity tracking, (ii) scene-level RAG strengthens factual accuracy, (iii) style profiling enforces persona

distinctiveness, and (iv) their integration within Fic2Bot yields additive improvements in persona consistency and factual grounding across diverse genres.

5.7.2. Efficiency Analysis

Table 11 presents the results of the efficiency analysis conducted as described in Section 4.5.2.

Table 11. Latency and resource analysis across novels. Build denotes the time required to build chatbots, measured in seconds.

Novel	Latency			Resource		
	Build (s)	**p50**	**p95**	**Token (In)**	**Token (Out)**	**Cost (USD)**
Winnie-the-Pooh	-	8.72	15.61	44,182.61	37.93	0.0556
Winnie-the-Pooh+Fic2Bot	4219.88	32.43	37.41	1852.50	299.80	0.0053
Off the Record	-	4.51	14.08	50,001.85	39.18	0.0629
Off the Record+Fic2Bot	6587.48	14.31	19.97	2149.02	22.95	0.0029
And What Can We Offer You Tonight	-	7.10	16.90	40,289	40	0.0508
And What Can We Offer You Tonight+Fic2Bot	5835.21	16.20	20.50	2654	30	0.0036

For *Winnie-the-Pooh*, the response latency was the longest, with a median (p50) of 32.43 s and a 95th percentile (p95) of 37.41 s. This latency can be attributed to the generation of output tokens more than ten times longer than those of the other two novels. The verbose and descriptive speaking style of the character led to lengthy outputs, producing more detailed responses but also incurring the highest cost (USD 0.0053).

For *Off the Record*, the response latency was p50 14.31 s and p95 19.97 s, with a relatively small gap between the two, indicating stable response times. The average input and output tokens were 2149.02 and 22.95, respectively. The character's direct and concise speaking style kept the outputs short, resulting in efficient operation at the lowest cost (USD 0.0029).

For *And What Can We Offer You Tonight*, the response latency was p50 16.20 s and p95 20.50 s, remaining within a narrow range. The average input and output tokens were 2654 and 30, respectively, and the character's compact and poetic style contributed to shorter outputs, resulting in a relatively low cost (USD 0.0036).

Among the three novels, *Off the Record* required the longest chatbot building time, approximately 110 min. This appears to stem from the frequent scene transitions characteristic of modern drama, which likely imposed additional overhead in the scene-level embedding process. In contrast, children's literature such as *Winnie-the-Pooh* has a simpler scene structure, resulting in shorter build times. In between these extremes, *And What Can We Offer You Tonight* exhibited an intermediate build time, positioned between the other two novels. Since chatbots are built only once when a novel is ingested, such differences do not translate into significant burdens for real-time operation. Finally, baseline, which relies on the entire text without additional retrieval or compression, exhibited shorter response latencies, with p50 and p95 ranging from a few to ten seconds. However, the number of input tokens per query increased excessively, resulting in high costs across all three novels, which could lead to exponential cost escalation in real-time operation.

6. Limitations

Several limitations remain in this study. First, the evaluation was confined to text-based novels, leaving applicability to multimodal narratives such as scripts or graphic novels untested.

Second, while prompt-based methods avoid additional fine-tuning, their heavy reliance on LLM prompting makes them vulnerable to performance variability across versions and providers. Notably, MultiLLM-Chatbot [43] evaluated five major LLM families across diverse metrics and domains, clearly demonstrating model-level differences. Although the overall architecture is still applicable, subtle variations in reasoning, style adherence, and retrieval alignment may cause inconsistencies, highlighting the need for systematic cross-LLM evaluation.

Third, human evaluation of subjective qualities such as fluency, empathy, and engagement was not included. Automated LLM-based judgments offer scalability but cannot fully capture human perception. Large-scale human studies will therefore be essential for validating persona consistency and understanding user experience in practice.

Finally, character style modeling relied on shallow linguistic features such as TF-IDF statistics, sentence length, and exclamation frequency. While informative, these features fail to capture deeper semantic, rhetorical, or narrative-level styles, limiting the richness of character expression. In addition, potential biases in sentiment analysis were not addressed in this study, which should be systematically examined in future work.

7. Conclusions

This paper presented Fic2Bot, an end-to-end framework that automatically transforms raw novel text into in-character chatbots by extracting character-specific linguistic and emotional profiles and integrating these with scene-grounded retrieval. Unlike conventional approaches, Fic2Bot achieves high performance without task-specific fine-tuning, making it both efficient and scalable.

Experiments across three novels demonstrated that the framework improves entity identification, retrieval precision, and persona consistency in multi-turn dialogues. These results highlight its ability to jointly enhance factual grounding and stylistic coherence. The significance of Fic2Bot lies not only in its empirical performance but also in its broader applicability, as it demonstrates scalability across diverse genres and narrative styles while adopting prompting strategies instead of additional training.

However, several promising directions remain for future research. First, generalizability should be validated beyond text-only novels, extending to diverse modalities and languages. Incorporating visual context, supported by recent advances in vision–language models [44] and multimodal RAG approaches [45], may enable richer narrative forms such as illustrated novels, scripts, and graphic narratives. Second, to mitigate dependence on LLM prompting, hybrid architectures that combine prompt-based methods with task-specific pretrained models should be explored, alongside systematic cross-LLM evaluations to assess robustness across model families. Third, large-scale human evaluations are needed to complement automated LLM-based judgments, particularly for subjective qualities such as fluency, empathy, and engagement. Finally, style modeling should move beyond shallow linguistic features (e.g., TF-IDF, sentence length) toward deeper semantic, rhetorical, and narrative-level modeling, while deployment feasibility must be further investigated with attention to efficiency, scalability, and lightweight implementations for real-world applications.

Author Contributions: Conceptualization, S.K., C.L. and S.J.; Funding acquisition, M.L.; Investigation, S.K., C.L. and S.J.; Methodology, S.K., C.L., S.J. and M.L.; Project administration, M.L.; Software, S.K., C.L. and S.J.; Supervision, M.L.; Writing—original draft, S.K., C.L. and S.J.; Writing—review & editing, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Sungshin Women's University Research Grant of 2025.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1

This prompt corresponds to the first stage of the MEI task.

Table A1. Example prompt and corresponding input/output for the Coreference Resolution task.

Prompt	You will receive a Text along with a list of Key Entities and their corresponding Cluster IDs as input. Your task is to perform Coreference Resolution on the provided text to categorize "each word belonging to a cluster" with its respective cluster id. Follow the format below to label a word with its cluster ID: word#cluster_id Please keep in mind: - Do not label general nouns, objects, or places. - Only tag names, pronouns, or descriptive phrases that refer to characters. - For example, tag both singular and plural references like: he, she, it, I, you, we, they, them, etc. - If a pronoun refers to multiple known characters, tag it with all relevant cluster IDs (e.g., they#pooh,#piglet). - Do not alter the content or order of the original text. - Ensure the output adheres to the specified format for easy parsing. - For characters not listed in Key Entities but appearing in the text, assign them the tag: #others **Key Entities:** {entity_description} **Text:** {text}
Input Example	**Key Entities:** "Winnie-the-Pooh": "#pooh", "Piglet": "#piglet" **Text:** Pooh went to the forest. He saw Piglet. They talked for hours.
Output Example	Pooh#pooh went to the forest. He#pooh saw Piglet#piglet. They#pooh,#piglet talked for hours.

Appendix A.2

Table A2. Prompt for expanding head words into full noun phrases.

Description	This prompt instructs the task of expanding an already tagged head word into its complete noun phrase, marking the full span accordingly.
Prompt	Any word marked with # is the head of a noun phrase. Expand this head by including its full span (e.g., determiners, adjectives, and modifiers). Do not add or remove content beyond bracketing. Wrap the full span using the format: (FULL SPAN)#ClusterID. Only bracket the expanded noun phrase, do not modify other parts of the sentence. If the head is a possessive determiner, ONLY bracket the determiner, NOT the noun it modifies, UNLESS the noun itself has a #ClusterID tag. **Text:** {text}
Input Example	Piglet#piglet wished very much that his#piglet Grandfather T. W.#others were there, instead of elsewhere.
Output Example	(Piglet)#piglet wished very much that (his Grandfather T. W.)#others were there, instead of elsewhere.

Appendix A.3

Table A3. Prompt for scene-based structuring after MEI.

Description	This prompt instructs the task of segmenting text data, after completing Major Entity Identification (MEI), into distinct scenes and converting them into a structured JSON format.
Prompt	You are a Scene Annotator. You will receive a story text input from {position}. Your task is to segment the story into distinct scenes based on any change in one or more of the following elements: - Location - Time - Characters present For each scene, extract the following structured information: - "Scene ID": e.g., "CH5_1", "CH5_2" - "Position": e.g., "CHAPTER_5" - "Speakers": list of speakers - "Scene": list of strings (full scene text including narration and dialogue) - "Dialogue": list of "Speaker: utterance" lines only Use consistent and full speaker names in both the "Speakers" list and the "Dialogue" lines. Do not use shortened or partial names. Refer to the following character list and ID mapping, and always use the full name when writing speaker names: {character_entities} Use this output format: [{"Scene ID": "CH5_1", "Position": "CHAPTER_5", "Speakers": ["Winnie-the-Pooh", "Christopher Robin"], "Scene": ["Winnie-the-Pooh walked through the forest...", ""Hello!" said Christopher Robin.", ""Hi!" said Winnie-the-Pooh."], "Dialogue": ["Christopher Robin: Hello!", "Winnie-the-Pooh: Hi!"]}] Now annotate the following chapter using SceneML: `{{chapter_text}}`
Input Example	**Position:** CHAPTER_5 **Character Entities:** "Winnie-the-Pooh": "pooh", "Christopher Robin" : "robin" **Text:** Winnie-the-Pooh walked through the forest. "Hello!" said Christopher Robin. "Hi!" said Winnie-the-Pooh.
Output Example	[{"Scene ID": "CH5_1", "Position": "CHAPTER_5", "Speakers": ["Winnie-the-Pooh", "Christopher Robin"], "Scene": ["Winnie-the-Pooh walked through the forest.", ""Hello!" said Christopher Robin.", ""Hi!" said Winnie-the-Pooh."], "Dialogue": ["Christopher Robin: Hello!", "Winnie-the-Pooh: Hi!"]}]

Appendix B

Table A4. Prompt for question rewriting.

Description	You are a helpful assistant that rewrites user questions to make them more effective for document retrieval systems.
Instructions	Please rewrite the question below to: • Remove ambiguity while preserving the original intent. • Keep the format as a question, not a declarative sentence. • Avoid guessing specific contexts not mentioned in the question. • Use concise and keyword-rich phrasing suitable for retrieval.
Original Question	{original_query}
Rewritten Question	{rewritten_query}

Instructs to rewrite the question while preserving its intent, maintaining the question format, and making it suitable for retrieval.

Appendix C

Appendix C provides the prompt templates used in Section 4.4 for persona consistency evaluation. They were applied to simulate user utterances in different scenarios and to guide GPT-4 in judging consistency. Each subsection presents the exact prompt texts given to the models.

Table A5. User prompt guidelines for each scenario.

ordinary conversation	You are simulating the USER in a two-person chat. Write the user's next single message in a neutral tone. Avoid role labels. Keep it short (1–2 sentences).
emotional conversation	You are simulating the USER in a two-person chat. Write the user's next single message with emotional intensity, responding to a conflict or disagreement. Avoid role labels. Keep it short (1–2 sentences).
fact-based conversation (first)	You are the USER inside a novel scene. Read the scene excerpt(s) below and write your first chat message as if you are physically present in that situation, addressing `{character_name}` directly. – The message should be a curious or probing question about what's happening *in the scene now*. – Use 1–2 short sentences. Avoid role labels. — Scene Excerpts — `{Scene}` Now write just the single user message (no quotes, no labels).
fact-based conversation (follow-up)	You are the USER inside the same ongoing novel scene. Stay fully in-world and speak as if you are physically here with the characters right now. Rules: · React directly to the assistant's last line in-character. · Use present tense; deictic words like here/now/this are fine if natural. · Keep it to 1–2 short sentences. · No role labels, no quotes, no stage directions, no meta-commentary. · If possible, reference a concrete detail from the assistant's line.

Table A6. Evaluation guidelines for character consistency analysis.

Role	You are an expert evaluator for character consistency analysis. You will be shown utterances from THREE different characters: Paisley, Hudson, and Leo. Each block contains multiple utterances representing that character's style. You will then be shown a [Target] block of utterances.
Task	1. Compare [Target] against ALL THREE character styles (Paisley, Hudson, Leo). 2. Based on similarity across four aspects, decide whether [Target] matches Paisley specifically. 3. Output ONLY in the format below.
Evaluation Criteria (1–5 scale)	1. Linguistic Style: Word choice patterns, punctuation usage (periods, ellipses, exclamations, questions), formality level, sentence length 2. Tone: Emotional attitude (formal/casual, warm/cold), confidence level (certain vs hesitant) 3. Vocabulary: Characteristic or idiosyncratic words/phrases, simplicity vs complexity, catchphrases or signature expressions 4. Emotional Expression: How the character expresses feelings (worry, joy, curiosity, etc.) 5. Dialogue Coherence: Appropriateness of the response to preceding user utterance, variability of expression (avoid repetition)
Guidelines	- Compare [Target] against ALL THREE characters, not just Paisley. - "Yes" only if [Target] clearly matches Paisley's stylistic and tonal patterns better than Hudson or Leo. - If evidence is weak or ambiguous, reflect lower confidence. - In the reason, cite specific linguistic evidence.
Output Format	"Linguistic Style": 1–5, "Tone": 1–5, "Vocabulary": 1–5, "Emotional Expression": 1–5, "Dialogue Coherence": 1–5, "average": (rounded average of the four scores, 1 decimal), "answer": "Yes|No", "confidence": "0–100%", "reason": "$\leq$50 words explaining your decision with reference to the criteria"

Appendix D

This Table presents the sentiment score distribution for the characters in *Off the Record*, derived from the emotion analysis described in the study.

Table A7. Provides the sentiment score distribution for the characters in *Off the Record* (%).

Character	1 Star	2 Stars	3 Stars	4 Stars	5 Stars
Andrea	27.78	8.33	22.22	5.56	36.11
Carrie	33.33	4.17	25.00	8.33	29.17
Dalton	33.33	16.67	33.33	0.00	16.67
Dorian	42.86	14.29	14.29	14.29	14.29
Hudson Owens	23.58	6.50	31.30	17.07	21.54
Kyla Langford	0.00	0.00	100.00	0.00	0.00
Leo Davis	20.00	5.00	15.00	5.00	55.00
Linus	28.57	14.29	14.29	14.29	28.57
Lucy	0.00	0.00	0.00	50.00	50.00
Luke	100.00	0.00	0.00	0.00	0.00
Moria Owens	50.00	50.00	0.00	0.00	0.00
Paisley McConkie	29.50	10.25	26.09	9.94	24.22
Phil Jenkins	25.00	0.00	0.00	0.00	75.00
Prescott	0.00	0.00	66.67	0.00	33.33
Simone Blake	29.76	7.14	27.38	11.90	23.81
Stan	40.00	20.00	20.00	0.00	20.00
Tina	25.00	12.50	0.00	25.00	37.50
Ralph	0.00	0.00	0.00	0.00	100.00

Appendix E

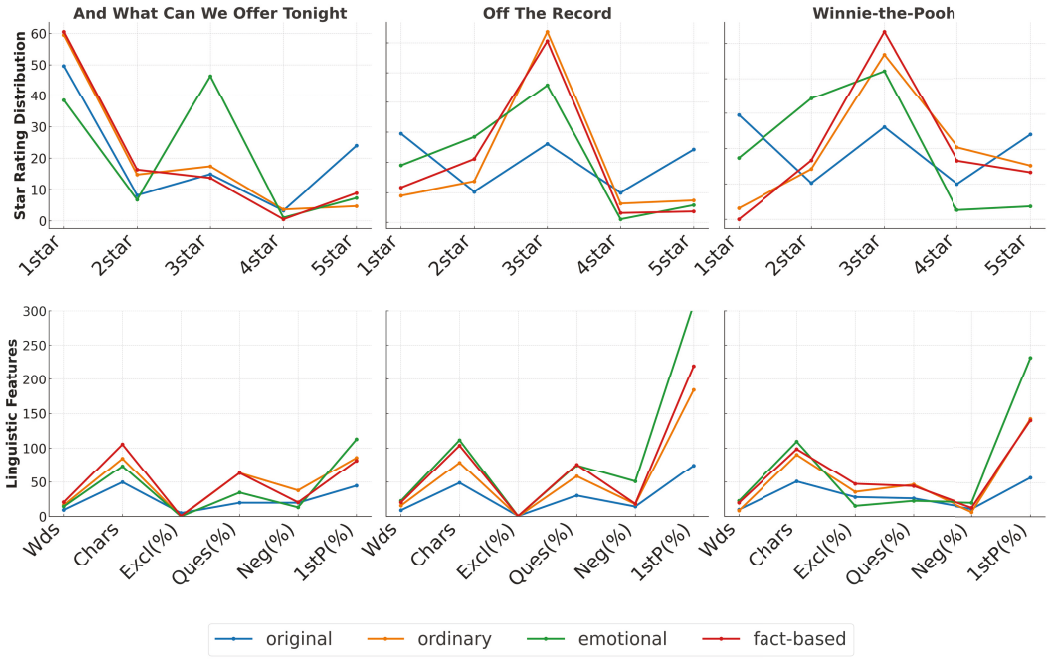


Figure A1. Comparative Analysis of Star Ratings and Linguistic Features Across Conversation Types. Abbreviations: Wds = Avg. Words; Chars = Avg. Characters; Excl = Exclamation; Ques = Question; Neg = Negation; 1stP = First-Person.

References

1. Mardiana, N.; Da, R.O. The Phenomenon of Use of Chatbot Character AI by ARMY-BTS. In Proceedings of the International Seminar Enrichment of Career by Knowledge of Language and Literature, Tokyo, Japan, 23–25 November 2024; pp. 65–74.
2. Han, S.; Kim, B.; Yoo, J.Y.; Seo, S.; Kim, S.; Erdenee, E.; Chang, B. Meet your favorite character: Open-domain chatbot mimicking fictional characters with only a few utterances. *arXiv* **2022**, arXiv:2204.10825. [CrossRef]

3. Wang, X.; Wang, H.; Zhang, Y.; Yuan, X.; Xu, R.; Huang, J.-t.; Yuan, S.; Guo, H.; Chen, J.; Zhou, S. Coser: Coordinating llm-based persona simulation of established roles. *arXiv* **2025**, arXiv:2502.09082.
4. Character.AI. Available online: https://character.ai (accessed on 9 July 2025).
5. Liu, Y.; Wei, W.; Liu, J.; Mao, X.; Fang, R.; Chen, D. Improving personality consistency in conversation by persona extending. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–22 October 2022; pp. 1350–1359.
6. Xue, B.; Wang, W.; Wang, H.; Mi, F.; Wang, R.; Wang, Y.; Shang, L.; Jiang, X.; Liu, Q.; Wong, K.-F. Improving Factual Consistency for Knowledge-Grounded Dialogue Systems via Knowledge Enhancement and Alignment. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; pp. 7829–7844.
7. Kovacevic, N.; Boschung, T.; Holz, C.; Gross, M.; Wampfler, R. Chatbots with attitude: Enhancing chatbot interactions through dynamic personality infusion. In Proceedings of the 6th ACM Conference on Conversational User Interfaces, Waterloo, ON, Canada, 8–10 July 2024; pp. 1–16.
8. Wang, Y.; Leung, J.; Shen, Z. RoleRAG: Enhancing LLM Role-Playing via Graph Guided Retrieval. *arXiv* **2025**, arXiv:2505.18541.
9. Liu, D.; Wu, Z.; Song, D.; Huang, H. A Persona-Aware LLM-Enhanced Framework for Multi-Session Personalized Dialogue Generation. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, 27 July–1 August 2025; pp. 103–123.
10. Zhu, S.; Ma, T.; Rong, H.; Al-Nabhan, N. A Personalized Multi-Turn Generation-Based Chatbot with Various-Persona-Distribution Data. *Appl. Sci.* **2023**, *13*, 3122. [CrossRef]
11. Vishnubhotla, K.; Rudzicz, F.; Hirst, G.; Hammond, A. Improving Automatic Quotation Attribution in Literary Novels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023; pp. 737–746.
12. Wang, N.; Peng, Z.y.; Que, H.; Liu, J.; Zhou, W.; Wu, Y.; Guo, H.; Gan, R.; Ni, Z.; Yang, J.; et al. RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 11–16 August 2024; pp. 14743–14777.
13. Hong, M.; Zhang, C.J.; Chen, C.; Lian, R.; Jiang, D. Dialogue Language Model with Large-Scale Persona Data Engineering. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, Albuquerque, NM, USA,29 April–4 May 2025; pp. 961–970.
14. Liu, Y.; Zhang, Y.; Patel, S.O.; Zhu, Z.; Guo, S. HonkaiChat: Companions from Anime that feel alive! *arXiv* **2025**, arXiv:2501.03277. [CrossRef]
15. Lee, O.; Joseph, K. A large-scale analysis of public-facing, community-built chatbots on Character. AI. *arXiv* **2025**, arXiv:2505.13354.
16. Lu, K.; Yu, B.; Zhou, C.; Zhou, J. Large Language Models are Superpositions of All Characters: Attaining Arbitrary Role-play via Self-Alignment. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024; pp. 7828–7840.
17. Li, C.; Leng, Z.; Yan, C.; Shen, J.; Wang, H.; Mi, W.; Fei, Y.; Feng, X.; Yan, S.; Wang, H. Chatharuhi: Reviving anime character in reality via large language model. *arXiv* **2023**, arXiv:2308.09597. [CrossRef]
18. Roesiger, I.; Schulz, S.; Reiter, N. Towards Coreference for Literary Text: Analyzing Domain-Specific Phenomena. In Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, Santa Fe, NM, USA, 25 August 2018; pp. 129–138.
19. Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N.F.; Peters, M.; Schmitz, M.; Zettlemoyer, L. AllenNLP: A Deep Semantic Natural Language Processing Platform. In Proceedings of Workshop for NLP Open Source Software (NLP-OSS), Melbourne, Australia, 15–20 July 2018; pp. 1–6.
20. Joshi, M.; Levy, O.; Zettlemoyer, L.; Weld, D. BERT for Coreference Resolution: Baselines and Analysis. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5803–5808.
21. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [CrossRef]
22. BookNLP. BookNLP: A Natural Language Processing Pipeline for Books. GitHub Repository. 2023. Available online: https://github.com/booknlp/booknlp (accessed on 9 August 2025).
23. Bamman, D.; Lewke, O.; Mansoor, A. An Annotated Dataset of Coreference in English Literature. In Proceedings of the Twelfth Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 44–54.
24. Sundar, K.M.; Toshniwal, S.; Tapaswi, M.; Gandhi, V. Major Entity Identification: A Generalizable Alternative to Coreference Resolution. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; pp. 11679–11695.
25. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.

26. Fan, W.; Ding, Y.; Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.-S.; Li, Q. A survey on rag meeting llms: Towards retrieval-augmented large language models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; pp. 6491–6501.
27. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, H.; Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv* **2023**, arXiv:2312.10997.
28. Olsson, F. Beyond the Basics: Advanced Retrieval Techniques for RAG Systems: Assessing the Impact of Sentence-Window Retrieval and Auto-Merging Retrieval on the Performance of a RAG System in a Swedish Management Consulting Company; Master's Thesis, KTH, School of Electrical Engineering and Computer Science, Stockholm, Sweden, 2024; p. 53.
29. Swacha, J.; Gracel, M. Retrieval-Augmented Generation (RAG) Chatbots for Education: A Survey of Applications. *Appl. Sci.* **2025**, *15*, 4234. [CrossRef]
30. Akkiraju, R.; Xu, A.; Bora, D.; Yu, T.; An, L.; Seth, V.; Shukla, A.; Gundecha, P.; Mehta, H.; Jha, A. Facts about building retrieval augmented generation-based chatbots. *arXiv* **2024**, arXiv:2407.07858. [CrossRef]
31. Khan, U.A.; Khan, F.; Khan, E.; Hasnain, M.A.; Moinuddin, A.A. Towards Efficient Educational Chatbots: Benchmarking RAG Frameworks. In Proceedings of the 2025 International Conference on Emerging Technologies in Computing and Communication (ETCC), Bangalore, India, 26–27 June 2025; pp. 1–6.
32. Gargari, O.K.; Habibi, G. Enhancing medical AI with retrieval-augmented generation: A mini narrative review. *Digit. Health* **2025**, *11*, 20552076251337177. [CrossRef]
33. Khasanova Zafar kizi, M.; Suh, Y. Design and Performance Evaluation of LLM-Based RAG Pipelines for Chatbot Services in International Student Admissions. *Electronics* **2025**, *14*, 3095. [CrossRef]
34. Alrashid, T.; Gaizauskas, R.J. A Pilot Study on Annotating Scenes in Narrative Text using SceneML. In Proceedings of the Text2Story Workshop at ECIR, Online, 1 April 2021; pp. 7–14.
35. Zeng, N.; Hou, H.; Yu, F.R.; Shi, S.; He, Y.T. SceneRAG: Scene-level Retrieval-Augmented Generation for Video Understanding. *arXiv* **2025**, arXiv:2506.07600.
36. Lee, J.; Chen, F.; Dua, S.; Cer, D.; Shanbhogue, M.; Naim, I.; Ábrego, G.H.; Li, Z.; Chen, K.; Vera, H.S. Gemini embedding: Generalizable embeddings from gemini. *arXiv* **2025**, arXiv:2503.07891. [CrossRef]
37. FAISS. Available online: https://faiss.ai/ (accessed on 8 August 2025).
38. BAAI. Bge-Reranker-Base. Hugging Face. 2023. Available online: https://huggingface.co/BAAI/bge-reranker-base (accessed on 9 August 2025).
39. NLPTown. BERT Base Multilingual Uncased Sentiment. Available online: https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment (accessed on 8 August 2025).
40. Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; Niu, S. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In Proceedings of the Eighth International Joint Conference on Natural Language Processing, Taipei, Taiwan, 27 November–1 December 2017; Volume 1: Long Papers; pp. 986–995.
41. Chen, H.Y.; Zhou, E.; Choi, J.D. Robust coreference resolution and entity linking on dialogues: Character identification on tv show transcripts. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 216–225.
42. Zhou, E.; Choi, J.D. They exist! introducing plural mentions to coreference resolution and entity linking. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 24–34.
43. Chakraborty, S.; Chowdhury, R.; Shuvo, S.R.; Chatterjee, R.; Roy, S. A scalable framework for evaluating multiple language models through cross-domain generation and hallucination detection. *Sci. Rep.* **2025**, *15*, 29981. [CrossRef]
44. Yu, X.; Yoo, S.; Lin, Y. Clipceil: Domain generalization through clip via channel refinement and image-text alignment. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 4267–4294.
45. Joshi, P.; Gupta, A.; Kumar, P.; Sisodia, M. Robust Multi Model RAG Pipeline for Documents Containing Text, Table & Images. In Proceedings of the 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Rajkot, India, 27–29 June 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 993–999.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Frequency-Aware Residual Networks with Masked Dual-Path Convolution for Image Classification

Jisoo Baik, Youngbeom Jung * and Kyujoong Lee *

School of AI Convergence, Sungshin Women's University, Seongbuk-gu, Seoul 02844, Republic of Korea; 20211344@sungshin.ac.kr

* Correspondence: ybjung@sungshin.ac.kr (Y.J.); kyujoonglee@sungshin.ac.kr (K.L.)

Abstract: Recent deep learning-based models have demonstrated outstanding performance in various computer vision tasks, including image classification. Among them, ResNet is a representative model that achieves high accuracy by utilizing a deep network architecture. However, as the depth of ResNet increases, the accumulated computational cost leads to higher overall FLOPs, which in turn results in increased computational burden and processing time. To address this issue, this paper proposes a novel model architecture that improves computational efficiency while maintaining accuracy. The proposed model reduces the total computational cost by separating the input image into high-frequency and low-frequency regions and applying convolution operations specialized to each region. Experimental results show that the proposed method reduces FLOPs by up to 29% compared to the original ResNet while maintaining competitive classification performance. In addition, by incorporating depthwise convolution instead of standard convolution, the model achieves an average reduction of 14.3% in operations compared to the standard multi-path model while reducing top-1 accuracy by only 0.18%.

Keywords: high/low frequency; depthwise convolution; image classification

1. Introduction

Image classification has long been a central task in computer vision, driving advancements across diverse applications such as autonomous driving, medical diagnostics [1], and security surveillance [2]. With the emergence of deep learning, convolutional neural networks (CNNs) have become the dominant architecture for achieving state-of-the-art (SOTA) performance. Models such as AlexNet [3], VGG [4], and particularly ResNet [5] have significantly improved classification accuracy by increasing network depth and incorporating architectural innovations like residual connections [5].

However, these accuracy gains come at the cost of higher computational complexity. Deep CNNs require substantial floating-point operations (FLOPs), leading to increased learning time, memory usage, and energy consumption. This limits their usage on resource-constrained platforms such as mobile devices or embedded systems [6,7]. To alleviate this burden, several approaches have been proposed: pruning [8], quantization [9], and efficient network designs like MobileNet [10] and ShuffleNet [11], which employ techniques such as depthwise separable convolutions and grouped convolutions.

Despite these advances, most models still apply convolutions uniformly across the entire image, regardless of local spatial characteristics. In real-world images, information is not evenly distributed. High-frequency regions tend to carry richer content than low-frequency regions. This observation has motivated a work that aims to separate convolution calculations based on decision masks [12].

In this paper, we propose a frequency-aware, multi-path convolutional architecture that dynamically adapts its operations based on the frequency composition of input images. By generating a binary mask that separates high-frequency and low-frequency regions, the model applies full standard convolution to information-rich areas and lightweight and depthwise separable convolution to smoother regions. This selective approach significantly reduces redundant computation while preserving key features for classification. We further enhance this mechanism by introducing a morphological dilation step to capture contextual neighborhoods around high-frequency regions, improving the model's receptive coverage without excessive computational cost.

Experiments on the CIFAR-10 dataset demonstrate that our method reduces computational cost by up to 29% compared to baseline ResNet models, while maintaining or slightly decreasing classification accuracy. These results highlight the effectiveness of combining frequency-based spatial analysis with efficient convolutional strategies for resource-constrained deep learning.

2. Related Works

Deep convolutional neural networks (CNNs) have led to substantial advancements in both image classification and image restoration tasks. In image classification, creative architectures such as VGG [4] and ResNet [5] have demonstrated that deeper networks with residual connections significantly improve performance. However, this comes at the cost of increased computational complexity, which has motivated a line of research into efficient CNN designs, including MobileNet [10] and ShuffleNet [11], which use depthwise separable and group convolutions to reduce FLOPs.

The most closely related work to our approach is ASCNN [12], which introduces an area-specific convolutional strategy for super-resolution by generating a binary decision mask that separates high-frequency and low-frequency regions. High-frequency regions are processed using full-capacity convolutions, while low-frequency areas are handled by lightweight convolutions, effectively reducing FLOPs. Furthermore, morphological dilation is used to expand the receptive context of high-frequency regions to improve restoration quality.

Inspired by ASCNN, our work adapts this frequency-aware design philosophy to the domain of image classification. Additionally, we extend the efficiency further by incorporating depthwise separable convolutions into the low-frequency path and optimizing mask through threshold control morphological dilation. This is the first work to integrate pixel-wise frequency-aware masking with dynamic convolution routing in the context of image classification. Unlike ASCNN [12], which was developed for single image super-resolution, our model is specifically designed for image classification. Moreover, our approach further improves efficiency by incorporating depthwise separable convolution into the low-frequency path. This extension highlights the novelty of our work, as it not only adapts ASCNN's core insight to a new domain but also introduces an additional mechanism to achieve greater FLOP reduction.

3. Proposed Method

This section provides an explanation of the core concepts underlying this study, including frequency-based mask generation, multi-path conventional convolution operation, and depthwise convolution.

The model adopts a ResNet-style backbone with three residual stages and replaces standard residual blocks with frequency-aware blocks. As illustrated in Figure 1, a binary mask is generated from the input based on local variation and thresholding, then optionally dilated. This mask is used to split feature maps into high- and low-frequency regions,

which are processed separately using a standard 3×3 convolution and a lightweight $3 \times 3 \rightarrow 1 \times 1$ bottleneck, respectively. The outputs are fused and passed through batch normalization and activation. After the final stage, global average pooling and a linear layer produce the final prediction. Key parameters include network depth, channel width, threshold theta, reduction ratio r, and dilation size d.

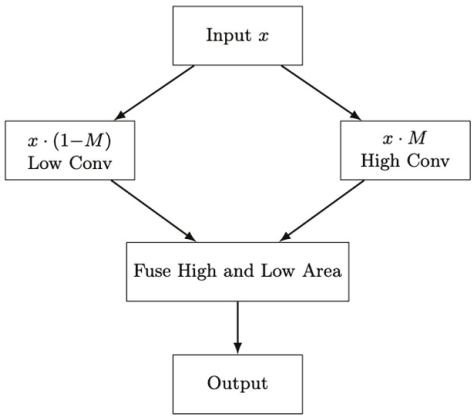


Figure 1. Minimal ASConvBlock diagram. Input is split by mask M, processed via two paths, and fused.

3.1. Frequency-Based Decision Mask

In conventional deep learning-based image classification models, convolution operations are typically applied uniformly across the entire input image or the resulting feature maps from the previous layers. However, real-world images contain distinct high-frequency and low-frequency regions, and accounting for this characteristic can lead to more computationally efficient operations. Notably, high-frequency regions tend to carry richer information than low-frequency ones, making it advantageous to focus computations more heavily on those areas. To achieve computational efficiency, the proposed method introduces a mechanism for generating a binary mask that separates high-frequency (HF) and low-frequency (LF) regions. This mask guides the convolution process to selectively apply operations based on the spatial frequency characteristics of the input. The mask generation procedure is as follows. First average pooling is applied to the input image to measure local variations. This operation effectively produces a blurred version of the input by retaining mainly low-frequency components. Then the absolute difference between the original input and the blurred image is computed. A pixel at location (i, j) is considered part of high-frequency region if the magnitude of this difference exceeds a predefined threshold θ, which serves as a hyperparameter. The resulting binary mask is later utilized during the conventional operation to differentiate between the HF and LF regions. The following Equation (1) compactly defines the mask-generation procedure.

$$M(x,y) = \begin{cases} 1, & if\ |I(x,y) - AvgPool(I)(x,y)| > \theta \\ 0, & otherwise \end{cases} \tag{1}$$

To illustrate the proposed mask generation process, Figure 2 presents a visual example using a 256×256 image of a butterfly. The top row shows the original image and its blurred counterpart obtained via average pooling, which retains only low-frequency components. The bottom row visualizes the absolute pixel-wise difference between the original and the blurred image, and the resulting binary mask that identifies high-frequency regions. The binary mask is computed by applying a threshold θ to the difference map. Specifically, pixels with local variation exceeding θ are assigned a value of 1 (high-frequency), while others are set to 0 (low-frequency). As θ increases, the condition for identifying high-frequency regions becomes stricter, resulting in a lower proportion of pixels labeled as 1,

thus producing a sparser mask. Conversely, decreasing θ relaxes the condition, yielding a denser mask with more pixels included as high-frequency regions. This frequency-aware masking mechanism enables the model to focus computational resources on more informative regions of the image, which are typically rich in sharp edges, textures, object boundaries and fine details. For clarity, the binary mask M is generated independently for each input image at the beginning of the forward pass. Masks are not pre-computed for the dataset nor shared across different images.



Figure 2. High-frequency mask generation process. The mask highlights regions where the absolute local variation exceeds the threshold $\theta = 0.072$. The threshold value was empirically chosen for effective visualization.

3.2. Multi-Path Convolution Block

To enhance computational efficiency while preserving representational capacity, we propose a multi-path convolutional block that selectively applies different convolution operations based on the frequency characteristics of the input. Specifically, the binary mask $M \in \{0, 1\}^{H \times W}$ generated from the input image guides the routing of the computation. The input feature map $X \in R^{Cin \times H \times W}$ is split into two separate components. For each spatial location, the model applies a full convolution path $Conv_{high}$, if $M = 1$ (i.e., high-frequency region), and a lightweight convolution path $Conv_{low}$ if $M = 0$ (i.e., low-frequency region). This process is shown in (2). Below, $\odot$ denotes element-wise multiplication.

$$X_{high} = X \odot M, \quad X_{low} = X \odot (1 - M) \tag{2}$$

In the high-frequency part, a standard $k \times k$ convolution is applied to X_{high}, maintaining the full output dimensionality C_{out}. For the low-frequency path, a computationally efficient strategy is employed: the number of channels is first reduced to C_{out}/r via a $k \times k$ convolution and then restored to C_{out} using a 1×1 convolution which is shown in (3).

$$Y_{high} = Conv_{high}\left(X_{high}\right), \quad Y_{low} = Conv_{1\times1}(Conv_{k\times k}(X_{low})) \tag{3}$$

Finally, the outputs of both paths are aggregated. This fusion effectively combines rich spatial detail from high-frequency regions with efficient approximations for low-frequency areas.

$$Y = Y_{high} + Y_{low} \tag{4}$$

By conditionally allocating computational resources in this manner, the model focuses capacity on information-rich (high-frequency) regions while maintaining efficiency in low-frequency areas. For additional information, the mask M is generated once from the input image and then reused throughout the network. At each stage transition, where the spatial resolution of the feature maps is reduced, the mask is down-sampled by a factor of 2 using average pooling. This operation converts the originally binary mask into a soft mask while maintaining its structural information. Inside each block, if the spatial resolution of M does not match that of the feature maps, the mask is resized via nearest-neighbor interpolation. This ensures consistent application of M to gate the high- and low-frequency paths across all stages.

3.3. Depthwise Convolution

While the proposed frequency-aware convolution block already reduces computational cost by applying lightweight convolutions to low-frequency regions, we further optimize this design by replacing standard convolutions in the low-frequency path with depthwise separable convolutions. This adjustment significantly reduces the number of floating-point operations (FLOPs). Therefore, replacing standard convolutions with depthwise convolutions in these early stages results in substantial computational savings. In particular, the low-frequency component X_{low} is processed using a depthwise convolution followed by a pointwise projection. The depthwise convolution performs spatial filtering independently for each input channel, without mixing information across channels. Formally, this operation applies C_{in} separate $k \times k$ kernels to C_{in} input channels, yielding an intermediate feature map of the same shape as shown in (5).

$$X'_{low} = DepthwiseConv_{k\times k}(X_{low}), withgroups = C_{in} \tag{5}$$

Subsequently, a 1×1 pointwise convolution aggregates and projects the channels into the target output dimension C_{out}. Which is shown in (6).

$$Y_{low} = Conv_{1\times 1}(X'_{low}) \tag{6}$$

As shown in Equation (4), the final output of the block remains the sum of the high-frequency and low-frequency outputs. By applying the depthwise separable convolution only to the low-frequency path we significantly reduce the computation cost without sacrificing high-frequency regions. This is particularly beneficial in deeper stages of the network or when scaling to higher-resolution inputs.

3.4. Morphological Dilation

To further refine the binary mask used for frequency-aware convolution, we apply a morphological dilation operation, which is a technique in image morphology [13], by using max pooling. Max pooling is equivalent to standard morphological dilation with a square kernel. This choice yields the same effect while being computationally efficient and natively supported in deep-learning frameworks like PyTorch 2.6.0. Specifically, after the high-frequency mask $M \in \{0, 1\}^{H\times W}$ is initially computed based on local variation, we perform the following update when the dilation parameter $d > 0$. This method is shown in (7).

$$M_{dilation} = MaxPool2D(M, kernel_size = d, stride = 1, padding = (d-1)/2) \tag{7}$$

This operation effectively expands the high-frequency regions by including their local neighborhoods. Although increasing the dilation parameter does not lead to a reduction in FLOPs—since more pixels are retained as high-frequency—it serves a different purpose:

allowing the model to consider not just the high-frequency points themselves, but also the nearby pixels that contribute to understanding them. In natural images, high-frequency areas such as edges and boundaries often contain critical semantic information. By dilating the mask, we ensure that not only the exact high-frequency pixels but also their surrounding pixels are processed with full convolution. This provides a balance between saving computation and preserving important image features. Empirically, we observe that larger dilation values (e.g., $d = 7$) yield better classification accuracy without significantly increasing computational cost. This suggests that the expanded receptive context of high-frequency zones plays a valuable role in preserving meaningful information. To further clarify the role of dilation in the frequency-aware masking mechanism, Figure 3 visualizes the change in binary masks across increasing dilation values. As seen in the figure, setting the dilation parameter to higher values (e.g., $d = 5$ or $d = 7$) results in the thickening of high-frequency regions, effectively expanding the mask boundaries. This morphological dilation includes surrounding pixels that are semantically related to structures such as edges and textures.

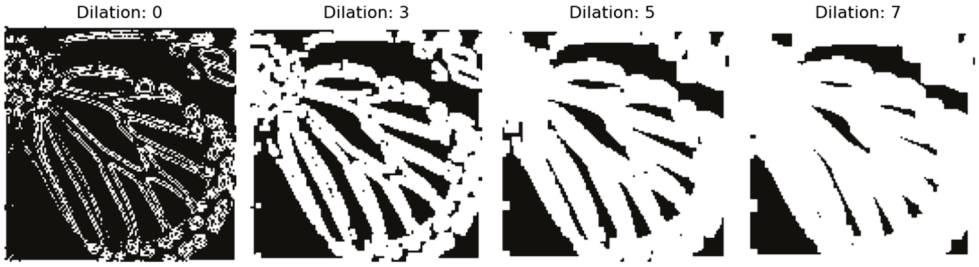


Figure 3. Visual comparison of frequency-based binary masks with varying dilation values ($d = 0, 3, 5, 7$).

4. Experiments

All models were implemented in PyTorch 2.6.0 and trained on a single NVIDIA RTX 3060 GPU. We used the CIFAR-10 dataset for both training and testing. We trained with SGD (momentum 0.9), initial learning rate 0.1, weight decay 5×10^{-4}, a cosine-annealing schedule over 150 epochs, and batch size 256. Method-level hyperparameters are tuned by grid search over the mask threshold θ and dilation $d \in \{0, 3, 5, 7\}$. To mitigate overfitting, we apply Random Crop (32, padding = 4) and Random Horizontal Flip, weight decay, input normalization with CIFAR-10 statistics, and batch normalization.

4.1. Effect of Morphological Dilation

To investigate the impact of expanding the high-frequency regions identified by the binary mask, we conducted a series of experiments incorporating morphological dilation into the frequency-aware convolutional framework. The goal of this approach is to enhance the contextual understanding of principal regions by extending convolutional coverage beyond strict high-frequency boundaries. This allows the model to more effectively capture local structure without significantly increasing computational burden. Models were trained with three different dilation settings $d \in \{3, 5, 7\}$, while all other components remained fixed. Importantly, the low-frequency convolution path used standard convolutions, and depthwise operations were not applied in this setting. Each model was fine-tuned from a common baseline ResNet-32 model with three residual stages and standard residual blocks without frequency-aware masking or depthwise convolution trained with CIFAR-10 dataset, ensuring consistent initialization.

The results shown in Table 1 demonstrate that increasing dilation generally leads to improved accuracy across varying MFLOPs (While the reported MFLOPs in Table 1 account only for convolutional layers, we additionally measured the cost of mask generalization.

For a batch of 256 images, the mask generation requires approximately 9.44 MFLOPs in total, which amounts to only about 5% of the full model (181 MFLOPs). This marginal contribution confirms that mask generation introduces negligible overhead relative to the dominant cost of convolutional operation.) Specifically, the model with $d = 7$ achieved top-1 accuracy of 94.98% at 168 MFLOPs—outperforming the baseline ResNet (94.80% at 181 MFLOPs) while using 13 MFLOPs fewer. Even under tighter computational budgets, such as 128 MFLOPs (this model is highlighted in bold in Table 1), the same model maintained 94.72% accuracy, only 0.08% below the baseline with 29% less computation. The reduction rate is calculated as $\{(181 - 128)/181\} \times 100\%$ which corresponds to a 29% decrease in computational complexity. This trend indicates that large dilation not only preserves important information along edges and object boundaries but also effectively extends the receptive field of full convolutions, resulting in consistent accuracy improvements at reduced FLOPs.

Table 1. Top-1 accuracy, MFLOPs and inference latency (ms per image) of models evaluated under various dilation settings without applying depthwise convolution.

		No Depthwise Convolution		
Dilation	**Threshold**	**FLOPs**	**Accuracy**	**Latency**
No Dilation (Baseline)	-	181M	94.80	1.957 ms
No Dilation	0.0107	188M	94.73	1.933 ms
	0.0181	168M	94.51	1.933 ms
	0.0283	148M	94.14	1.933 ms
	0.0469	128M	93.85	1.931 ms
3×3	0.0469	188M	94.60	1.995 ms
	0.0723	168M	94.73	1.995 ms
	0.1016	148M	94.68	1.977 ms
	0.1267	128M	94.21	1.992 ms
5×5	0.1016	188M	94.73	1.935 ms
	0.1211	168M	94.83	1.919 ms
	0.1450	148M	94.53	1.921 ms
	0.1631	128M	94.64	1.916 ms
7×7	0.1211	188M	94.95	1.935 ms
	0.1543	168M	94.98	1.917 ms
	0.1719	148M	94.84	1.917 ms
	0.2109	**128M**	**94.72**	**1.916 ms**

While Table 1 shows that larger dilation values consistently improve accuracy by extending the receptive field, Figure 4 reveals that this also leads to a steady increase in MFLOPs. The expansion of the binary mask at higher dilation levels causes high-cost convolutions to cover larger image regions, raising computational complexity. To address this, our design adjusts the threshold parameter θ to maintain a balanced mask size, thereby limiting unnecessary growth into high-frequency areas. This prevents the mask from becoming excessively thick and allows the model to capture richer contextual information without proportionally inflating computational cost. Consequently, the proposed approach achieves accuracy gains comparable to large-dilation configurations while preserving computational efficiency.

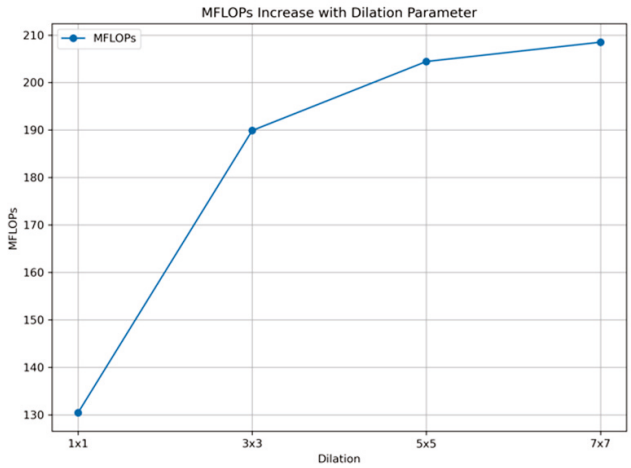


Figure 4. Trend of increasing MFLOPs with larger Dilation values. Threshold with 0.044.

4.2. Effect of Depthwise Convolution

Building upon the benefits of dilation, we further explored the effect of applying depthwise separable convolutions to the low-frequency path. Unlike standard convolution, which operates across both spatial and channel dimensions simultaneously, depthwise separable convolution splits this process into a spatially focused depthwise operation followed by a lightweight pointwise projection. This design drastically reduces FLOPs, especially beneficial in early network stages where feature maps are large.

Figure 5 clearly illustrates this advantage: across all threshold settings ($\theta \in \{0.016, 0.044, 0.072, 0.1\}$), models with depthwise convolution consistently require fewer operations than their standard-convolution counterparts. In the figure, the numerical labels placed above each point indicate the corresponding top-1 accuracy (%) for that configuration, allowing direct comparison of efficiency and performance. Relative to the standard multi-path model, the depthwise variant achieves an average MFLOP reduction of 17.5, corresponding to a 14.3% decrease in operations, while the corresponding average drop in top-1 accuracy remains at only 0.18%.

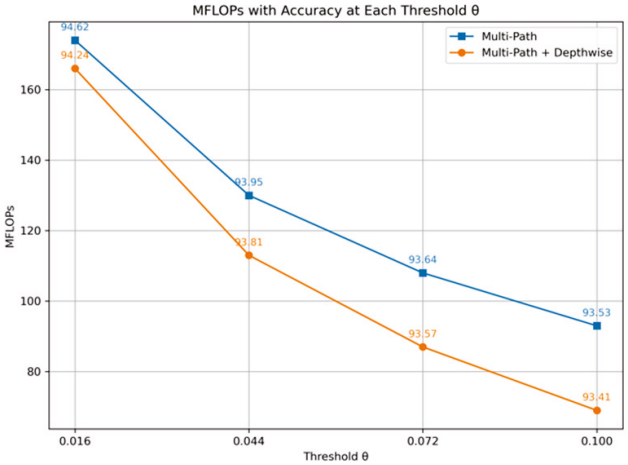


Figure 5. MFLOPs and accuracy reduction trend and as the threshold θ increases for different model configurations.

These results confirm that depthwise convolution delivers double-digit percentage reductions in FLOPs across a wide range of thresholds, while maintaining accuracy drop within 0.18% of the standard model. The minimal performance degradation suggests that

the reduced inter-channel interaction inherent to depthwise convolution is not harmful in the low-frequency path, where structural details are less critical.

When combined with morphological dilation in the high-frequency path, depthwise convolution forms a complementary mechanism: dilation expands the receptive field to capture contextual detail along edges, while depthwise convolution minimized computational redundancy in smooth regions. This synergy enables the proposed architecture to maintain competitive top-1 accuracy while significantly lowering computational demand, validating the use of depthwise convolution as a core element of frequency-aware design for resource-constrained environment.

4.3. Additional Experiments

To further validate the effectiveness of the proposed dual-path and depthwise convolution design, we conducted additional experiments under the more challenging CIFAR-100 dataset. For consistency, the dilation parameter was fixed at 7, which had been identified in Table 1 as the most effective configuration in terms of accuracy–efficiency trade-off. All models were trained under the same conditions, using 200 epochs with SGD, a cosine learning rate schedule, with standard CIFAR augmentations.

The results summarized in Table 2 demonstrate that the proposed design consistently reduces FLOPs while maintaining nearly the same accuracy as the baseline ResNet. Here, the baseline refers to a standard ResNet trained on CIFAR-100 without incorporating dual-path routing or depthwise convolutions. The baseline model achieves 77.58% top-1 accuracy at 647.72 MFLOPs. While our best-performing dual-path and depthwise variant reaches 77.53% accuracy with only 546 MFLOPs (Our-546 Model), corresponding to a 15.7% reduction in FLOPs, computed as $(647.72 - 546)/647.72 \times 100\%$, with only 0.05% accuracy drop. This model is highlighted in bold in Table 2.

Table 2. Baseline ResNet and proposed dual-path + depthwise models on CIFAR-100.

Model	Threshold	FLOPs	Accuracy
Baseline	-	647.72M	77.58
Our-616 Model	0.1387	616M	77.54
Our-546 Model	**0.1582**	**546M**	**77.53**
Our-476 Model	0.1719	476M	77.41

This trend indicates that the dual-path design allows the model to selectively allocate expensive full convolutions only to critical high-frequency regions, while employing lightweight depthwise operations elsewhere. As a result, the network maintains rich representational capacity despite substantial reductions in FLOPs. Importantly, the ability to preserve baseline-level accuracy under the more complex CIFAR-100 dataset further highlights the generalizability and robustness of our approach.

We further investigated whether the proposed dual-path and depthwise convolution design provides benefits when applied to a lightweight backbone such as EfficientNet-B0.

As shown in Table 3, the baseline EfficientNet-B0 trained on CIFAR-10 achieves 91.56% top-1 accuracy with 30.76 MFLOPs. When incorporating our dual-path routing and depthwise convolution, the resulting model attains 91.52% accuracy while reducing the computational cost to 28.98 MFLOPs. In this setting, the dilation parameter was fixed to $d = 7$, consistent with the ResNet experiments described above, and the threshold θ was selected through a grid search, which identified $\theta = 0.002$ as the most effective configuration. This choice provides a 5.8% FLOP reduction with only a negligible accuracy drop of 0.04%.

These results demonstrate that the proposed design extends effectively to lightweight backbones, confirming its robustness across different network families.

Table 3. Baseline EfficientNet and proposed dual-path + depthwise models on CIFAR-10.

Model	FLOPs	Accuracy
EfficientNet-B0 (Baseline)	30.76M	91.56
EfficientNet-B0 (DualPath + Depthwise)	28.98M	91.52

5. Conclusions

This paper presents a frequency-aware, multi-path convolutional architecture that delivers competitive classification performance while substantially reducing computational cost. By decomposing input features into high- and low-frequency regions via a binary mask, our model dynamically applies full convolutional operations to information-rich areas and lightweight operations to smoother regions. This selective computation strategy preserves accuracy where it is most critical while minimizing redundant processing in less informative zones. The introduction of morphological dilation enhances the mask's effectiveness by expanding the spatial coverage of high-frequency regions, thereby capturing valuable contextual information around edges and fine details. Furthermore, incorporating depthwise separable convolutions in the low-frequency path reduces floating-point operations (FLOPs) without compromising accuracy-particularly in the early stages where feature maps are large.

Unlike conventional ResNet models, the proposed method leverages spatial-frequency priors to dynamically reallocate computational resources, enabling efficient deployment in resource-constrained environments such as edge devices and real-time applications without sacrificing classification performance. In our experiments, the proposed method reduced FLOPs by 29% compared to original ResNet with slightly reduced accuracy. Also, by replacing standard convolution with depthwise convolution in the low-frequency path achieved an additional average 14.3% reduction over the standard multi-path model with only a 0.18% drop in top-1 accuracy. Overall, our results demonstrate that a principled, frequency-aware design can effectively break the traditional trade-off between accuracy and efficiency in deep convolutional networks.

Author Contributions: Conceptualization, K.L.; Software, J.B.; Writing—Original Draft Preparation, J.B.; Writing—Review and Editing, Y.J. and K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study are publicly available. CIFAR-10 and CIFAR-100 can be accessed at https://www.cs.toronto.edu/~kriz/cifar.html and are also available via the torchvision.datasets library in PyTorch. URL accessed on 11 April 2024.

Acknowledgments: This work was supported by the Sungshin Women's University Research Grant of 2025 (Grand Recipient: Kyujoong Lee).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, Y.; Sixou, B.; Peyrin, F. A review of the deep learning methods for medical imager super resolution problems. *IRBM* **2021**, *42*, 120–133. [CrossRef]
2. Rasti, P.; Uiboupin, T.; Escalera, S.; Anbarjafari, G. Convolutional neural network super resolution for face recognition in surveillance monitoring. In Proceedings of the 9th International Conference, Articulated Motion and Deformable Objects, Palma de Mallorca, Spain, 13–15 July 2016; pp. 175–184.

3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
4. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
6. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both Weights and Connections for Efficient Neural Network. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
7. Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J.S. Efficient Processing of Deep Neural Networks. *Proc. IEEE* **2017**, *105*, 2295–2329. [CrossRef]
8. Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; Kautz, J. Pruning Convolutional Neural Networks for Resource Efficient Transfer Learning. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
9. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
10. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
11. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
12. Alao, H.; Kim, T.S.; Lee, K. Area-Specific Convolutional Neural Networks for Single Image Super-Resolution. *IEEE Access* **2022**, *10*, 104567–104576. [CrossRef]
13. Goyal, M. Morphological image processing. *Int. J. Comput. Sci. Telecommun.* **2011**, *2*, 59.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

A Short-Term User-Side Load Forecasting Method Based on the MCPO-VMD-FDFE Decomposition-Enhanced Framework

Yu Du [1,*], Jiaju Shi [2], Xun Dou [3] and Yu He [3]

[1] NARI Nanjing Control System Co., Ltd., Nanjing 211106, China
[2] School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China; sjj0810@stu.xjtu.edu.cn
[3] School of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing 211816, China; dxnjut@njtech.edu.cn (X.D.); heyu@njtech.edu.cn (Y.H.)
* Correspondence: duyu@sgepri.sgcc.com.cn

Abstract: With the transition of the energy structure and the continuous development of smart grids, short-term user-side load forecasting plays a key role in fine power dispatch and efficient system operation. However, existing parameter optimization methods lack multi-dimensional and physically interpretable fitness evaluation. They also fail to fully exploit frequency-domain features of decomposed modal components. These limitations reduce model accuracy and robustness in complex scenarios. To address this issue, this paper proposes a short-term user-side load forecasting method based on the MCPO-VMD-FDFE decomposition-enhanced framework. Firstly, a multi-dimensional fitness function is designed using indicators such as modal energy entropy and energy concentration. The Crested Porcupine Optimizer with Multidimensional Fitness Function (MCPO) algorithm is applied in VMD (Variational Mode Decomposition) to optimize the number of decomposition modes (K) and the penalty factor (α), thereby improving decomposition quality. Secondly, each IMF component obtained from VMD is analyzed by FFT. Key frequency components are selectively enhanced based on adaptive thresholds and weight coefficients to improve feature expression. Finally, a multi-scale convolution module is added to the PatchTST model to enhance its ability to capture local and multi-scale temporal features. The enhanced IMF components are fed into the improved model for prediction, and the final output is obtained by aggregating the results of all components. Experimental results show that the proposed method achieves the best performance on user-side load datasets for weekdays, Saturdays, and Sundays. The RMSE is reduced by 45.65% overall, confirming the effectiveness of the proposed approach in short-term user-side load forecasting tasks.

Keywords: optimization algorithm; modal decomposition; frequency-domain feature enhancement; multi-scale convolution; load forecasting

1. Introduction

Short-term user-side load forecasting serves as an essential tool for power system dispatching, operational optimization, and demand-side management [1–3]. It provides accurate load demand information to power utilities, enabling optimized generation planning, reduced operational costs, and improved system reliability and stability. However, with the large-scale integration of distributed energy resources, the volatility and nonlinearity of load sequences have increased significantly, making accurate forecasting more challenging [4–6]. Therefore, developing efficient and accurate short-term user-side load forecasting methods holds significant theoretical and practical value [7–9].

Traditional load forecasting methods mainly include time series analysis, regression analysis, and exponential smoothing. However, these methods often yield low prediction accuracy and fail to meet the growing demands of modern power grids. In contrast, machine learning and deep learning methods offer better capabilities in handling complex nonlinear relationships and large-scale data. Reference [10] proposed a day-ahead net load forecasting method based on iTransformer to address challenges posed by high penetration of renewable energy. This approach considers components such as photovoltaic generation, wind power, and active loads, demonstrating high accuracy and stable performance. Reference [11] introduced a short-term load forecasting method under electricity market conditions, incorporating real-time electricity prices. This method effectively captures the correlation between price and load, thereby improving prediction accuracy. To address the increasing diversity and randomness of loads in new power systems, Reference [12] proposed a multi-step short-term load forecasting method based on user group segmentation. The results show that the method significantly improves forecasting accuracy and stability across the entire user group. Reference [13] developed a short-term load forecasting model based on Bi-directional LSTM for multiple buildings. The model enhances the representation of unstable peaks and inter-building correlations. Experimental results show substantial improvements in RMSE and MAE compared to traditional LSTM. Reference [14] proposed a short-term forecasting model based on residual-stacked frequency attention to improve load forecasting accuracy in low-voltage distribution networks. This model highlights the main trend features of load signals and outperforms existing deep learning models on multiple datasets in terms of prediction accuracy. Reference [15] proposed TransformGraph, a Transformer-based model that integrates graph convolutional networks (GCNs) to improve electricity net load forecasting. By encoding correlations through GCN and capturing temporal dependencies with self-attention, the model demonstrated superior accuracy and stability compared with five baseline approaches on OPSD datasets from three countries. Reference [16] explored the application of one-dimensional CNNs based on modified Video Pixel Networks (VPNs) for short-term load forecasting, addressing the limited use of CNNs compared with LSTM and GRU in time series problems. Experiments on real-world datasets for 1-h-ahead and 24-h-ahead tasks showed that the proposed CNN model achieved the best overall performance, with 2.21% MAPE for 24-h predictions.

However, due to the large-scale integration of distributed energy resources, the nonlinearity and nonstationarity of load variations have intensified. Relying on a single forecasting model has clear limitations. Constructing a hybrid framework that combines decomposition and prediction can effectively improve forecasting accuracy and robustness while also reducing the risk of overfitting. Reference [17] developed an ensemble forecasting model based on optimized variational mode decomposition (VMD), an improved sand cat swarm algorithm, and a bi-directional long short-term memory (Bi-LSTM) network. The model introduces the beluga whale algorithm to optimize the number of decomposition layers and the penalty factor, leading to enhanced forecasting stability. Reference [18] proposed a hybrid forecasting model combining VMD and support vector regression (SVR), integrating a chaotic mapping mechanism and an improved grey wolf optimizer (CGWO) to optimize SVR parameters. This approach alleviates premature convergence and enhances the exploration ability of the solution space, thus improving forecasting accuracy. To enhance the accuracy of residential load forecasting, Reference [19] introduced a combined deep learning model that considers multi-scale electricity usage behavior of individual households. The model integrates backpropagation neural networks, extreme gradient boosting, and LSTM, achieving stable and accurate load forecasts. Reference [20] presented a combined short-term load forecasting model based on empirical mode decomposition (EMD), extended Kalman filtering, and kernel-based extreme

learning machines. Case studies on user-side microgrids demonstrate the model's accuracy, update stability, and computational efficiency. Reference [21] proposed a hybrid forecasting method for air-conditioning cooling loads, combining VMD and an improved least squares support vector machine (LSSVM). Whale optimization was employed for parameter tuning, enabling effective decomposition of load sequences. Results show the method provides high prediction accuracy and robustness. Reference [22] proposed a short-term load forecasting method combining ICEEMDAN-RLMD dual decomposition with a CrossInformer architecture. The approach employs intelligent optimization for feature selection, integrates multi-granularity patch inputs and multiple attention mechanisms to enhance sequence modeling, and applies SHAP to improve interpretability. Experimental results demonstrate superior accuracy and stability compared with traditional models. Reference [23] developed an ultra-short-term load forecasting model integrating IVMD–SGMD decomposition with BiLSTM enhanced by temporal pattern attention, optimized by INGO. A TCN-based error correction module further improved accuracy, and experiments on quarterly datasets confirmed the model's high precision and robustness. Despite these advances, two critical challenges remain. First, most optimization algorithms guiding modal decomposition lack multi-dimensional and physically interpretable fitness evaluation, which hinders the effective identification of modal components. Second, the processing of decomposed signals typically remains at the raw input level, without further exploration of their frequency-domain features.

To address the above issues, this paper proposes a short-term user-side load forecasting method based on the MCPO-VMD-FDFE decomposition-enhanced framework. The main contributions of this study are as follows:

1. A modal decomposition parameter optimization method driven by a multi-dimensional, physically interpretable fitness function is proposed. A fitness function is designed by integrating modal energy entropy, orthogonality, and dominant frequency concentration. Combined with the Crested Porcupine Optimizer with Multidimensional Fitness Function (MCPO), the method jointly optimizes the VMD parameters, significantly improving decomposition quality and physical interpretability.
2. A frequency-domain feature enhancement mechanism based on adaptive thresholding and weighted strategies is introduced. Key frequency components are selected using FFT spectral analysis and adaptive thresholding, followed by selective enhancement based on peak significance weighting. This approach effectively strengthens the frequency-domain expressiveness of the modal components.
3. An MSCPatchTST model is constructed by integrating a multi-scale convolution module with channel independence. While maintaining the channel independence advantage of PatchTST, the model introduces parallel 1D convolutions at different scales to extract multi-granularity local features. An adaptive fusion mechanism further enhances the model's ability to capture both local and multi-scale dynamic features of load sequences.

2. Methodology

The overall framework of the proposed method is illustrated in Figure 1. First, a multi-dimensional fitness function is constructed, and the MCPO algorithm is applied to jointly optimize the number of decomposition modes K and the penalty factor α in VMD. Second, fast Fourier transform (FFT) is used to perform spectral analysis on the VMD components. Key frequency components are selectively amplified based on adaptive thresholding and a weighted enhancement strategy, thereby improving their frequency-domain representation. Finally, the enhanced modal components are fed into a modified PatchTST model with a multi-scale convolution module. This allows the model to capture both local details and

dynamic features across different time scales. The final prediction result is obtained by aggregating the outputs of all modal components.

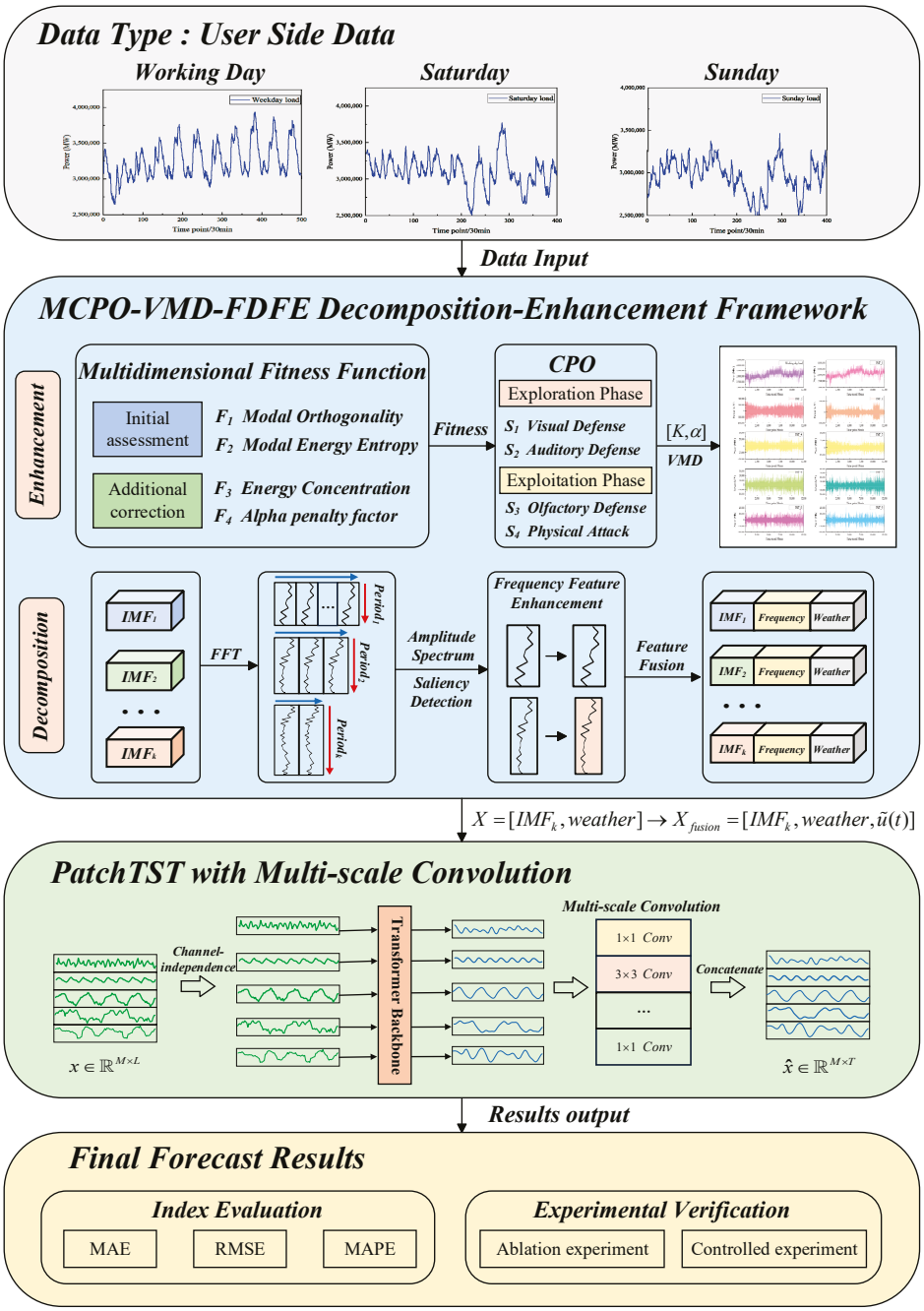


Figure 1. Overall framework.

2.1. MCPO-VMD-FDFE Decomposition-Enhanced Framework

To extract the multi-frequency features embedded in user-side load data and enhance the frequency-domain representation of intrinsic mode functions (IMFs), a comprehensive evaluation function is constructed by integrating modal energy entropy and energy concentration. This replaces the traditional single-objective fitness function. The Crowned Pigs Optimizer (CPO) is then used to optimize the number of decomposition modes K and the penalty factor α in VMD. Meanwhile, FFT is applied to perform spectral analysis on the IMF components. Key frequency components are selectively enhanced based on adaptive thresholds and weighting coefficients, thereby improving the feature expressiveness of the modal components.

2.1.1. VMD

Variational Mode Decomposition (VMD) is an adaptive signal decomposition method that can extract modal components at different frequency scales. It is effective in capturing the multi-scale characteristics and fluctuation patterns of load data. Essentially, VMD formulates a constrained variational problem, as expressed by the following equation:

$$\begin{cases} \min\limits_{\{u_k\}\{\omega_k\}} \left\{ \sum\limits_{k=1}^{k} ||\partial_t \left[\left(\delta(t) + \dfrac{\mathrm{j}}{\pi t} \right) u_k(t) \right] \mathrm{e}^{-\mathrm{j}\omega_k t}||_2^2 \right\} \\ \text{s.t.} \sum\limits_{k=1}^{K} u_k = f(t) \end{cases} \tag{1}$$

where k represents the total number of modes to be decomposed; ∂_t is the t-th order moment deviation, reflecting the time-domain characteristics of the signal; $\delta(t)$ denotes the Dirac distribution function, which ensures the localization of each mode in the time-frequency domain; u_k refers to the k-th intrinsic mode component obtained through decomposition; ω_k is the center frequency associated with each mode; $||.||$ indicates the number of elements computed under the L2 norm; and $f(t)$ is the original input signal to be decomposed.

By introducing a Lagrange multiplier and a quadratic penalty term, the constrained problem can be transformed into an unconstrained optimization problem.

$$\begin{cases} L(\{u_k\},\{\omega_k\},\lambda) = \alpha \sum\limits_{k=1}^{k} ||\partial_t \left[\left(\delta(t) + \dfrac{\mathrm{j}}{\pi t} \right) \odot u_k \right] \mathrm{e}^{-\mathrm{j}\omega_k t}||_2^2 + \\ ||f(t) - \sum\limits_{k=1}^{k} u_k||_2^2 + \lambda(t), f(t) - \sum\limits_{k=1}^{k} u_k(t) \end{cases} \tag{2}$$

where λ is the Lagrange multiplier; α is the penalty factor; and $\odot$ denotes the convolution operator.

2.1.2. MCPO

(1) CPO

CPO is a novel metaheuristic algorithm. Its flowchart is shown in Figure 2. The algorithm optimizes the objective function by simulating four defensive behaviors of crowned pigs when facing predators. A cyclic population contraction strategy is employed to periodically reduce the number of active individuals during the iteration process. This dynamic reduction in population size accelerates convergence while maintaining population diversity. The initialization process and the formula for cyclic population contraction are defined as follows:

$$\vec{\mathrm{X}_\mathrm{i}} = \vec{\mathrm{L}} + \vec{\mathrm{r}}(\vec{\mathrm{R}} - \vec{\mathrm{L}}), i = 1, 2 \ldots, n, \tag{3}$$

$$N = N_{min} + (N' - N_{min}) \left(1 - \left(\frac{t\%\frac{T_{max}}{T}}{\frac{T_{max}}{T}} \right) \right) \tag{4}$$

where n denotes the population size; $\vec{\mathrm{X}_\mathrm{i}}$ represents the i-th candidate solution in the search space; $\vec{\mathrm{L}}$ and $\vec{\mathrm{R}}$ are the upper and lower bounds of the search range, respectively; $\vec{\mathrm{r}}$ is a random number in the range [0, 1]; T is the predefined cycle period; t denotes the number of function evaluations; T_{max} is the maximum value of the evaluation function; % represents the remainder or modulo operator, and N_{min} is the minimum number of newly generated individuals in the population.

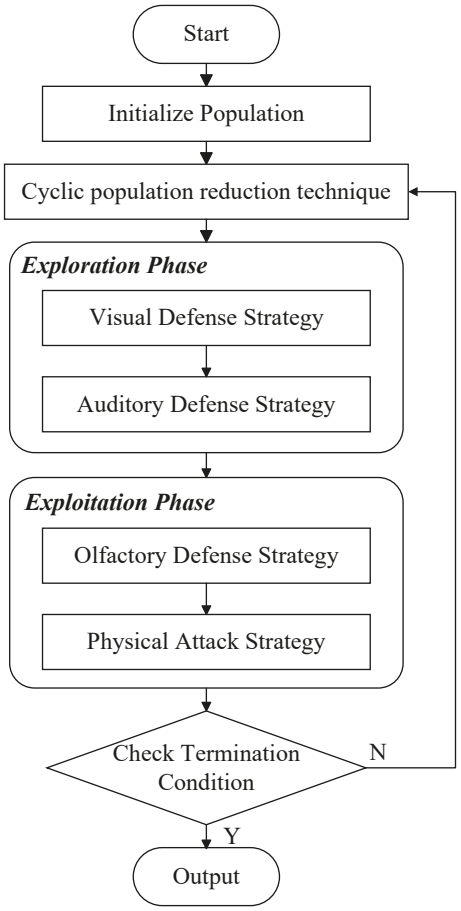


Figure 2. Flowchart of CPO.

During the search process, CPO divides the update procedure into two phases: exploration and exploitation. The exploration phase includes two strategies: (i) perturbation updates around the global best solution to increase solution diversity and (ii) convergence adjustment using adaptive scaling factors to regulate search speed. The exploitation phase also includes two strategies: (iii) local refinement through information exchange between individuals; (iv) local intensification using directional and defense factors to accelerate convergence and improve solution accuracy.

The mathematical formulations of the four update strategies are given in Equations (5)–(8). These mechanisms work in a complementary manner, providing a balance between global exploration and local exploitation, and enhancing the algorithm's performance on complex optimization problems.

$$\vec{x}_i^{t+1} = \vec{x}_i^{t} + \tau_1 \times \left| 2 \times \tau_2 \times \vec{x}_{CP}^{t} - \vec{y}_i^{t} \right| \tag{5}$$

where $\vec{x}_{CP}^{t}$ denotes the current optimal solution of the evaluation function; $\vec{y}_i^{t}$ represents the position of the predator at iteration time t; τ_1 is a random value following a normal distribution; and τ_2 is a random number within the range [0, 1]

$$\vec{x}_i^{t+1} = \vec{x}_{CP}^{t} + (\alpha(1-\tau_4)+\tau_4) \times \left(\delta \times \vec{x}_{CP}^{t} - \vec{x}_i^{t} \right) - \tau_5 \times \delta \times \gamma_t \times \vec{F}_i^{t} \tag{6}$$

where $\vec{x}_{CP}^{t}$ denotes the obtained optimal solution; τ_4 is a random value within the range [1, 4]; α is the convergence rate factor; and $\vec{F}_i^{t}$ represents the average force acting on the crested porcupine (CP).

$$\vec{x}_i^{t+1} = \left(1 - \vec{U}_1\right) \times \vec{x}_i^{t} + \vec{U}_1 \times \left(\vec{y} + \tau_3 \times \left(\vec{x}_{r1}^{t} - \vec{x}_{r2}^{t}\right)\right) \tag{7}$$

where $\vec{x}_{r1}^{t}$ and $\vec{x}_{r2}^{t}$ are two random integers in the range [1, N], and τ_3 is a random number within the range [0, 1].

$$\vec{x}_i^{t+1} = \left(1 - \vec{R}_1\right) \times \vec{x}_i^{t} + \vec{R}_1 \times \left(\vec{x}_{r1}^{t} + S_i^t \times \left(\vec{x}_{r2}^{t} - \vec{x}_{r3}^{t}\right) - \tau_3 \times \delta \times \gamma_t \times S_i^t\right) \tag{8}$$

where $\vec{x}_i^{t}$ denotes the position of the i-th individual at iteration t; γ_t is the defense factor; $r3$ is a random value within the range [1, N]; δ controls the search direction; τ_3 is a random number in the range [0, 1]; and S_i^t represents the odor diffusion factor.

(2) Multi-dimensional Fitness Function

To overcome the limitations of single-objective fitness functions in the variational mode decomposition (VMD) of power load signals, this study proposes a multi-dimensional fitness function by integrating modal energy entropy, orthogonality, and dominant frequency distinguishability. An Alpha-based penalty factor is also introduced. A modified MCPO optimization algorithm based on this multi-dimensional fitness function is developed to enhance the decomposition of multi-scale features in both time and frequency domains. This approach ensures that the resulting modes retain clear physical interpretability and structural rationality while significantly improving the stability and robustness of the decomposition process. The specific formulation of the multi-dimensional fitness function is as follows:

(a) Modal Energy Entropy

Modal energy entropy is used to measure the concentration of energy distribution across the decomposed components. A lower entropy value indicates more concentrated energy, suggesting better decomposition performance. In contrast, a higher entropy value may imply modal redundancy and lower decomposition quality.

$$\vec{x}_i^{t+1} = \left(1 - \vec{R}_1\right) \times \vec{x}_i^{t} + \vec{R}_1 \times \left(\vec{x}_{r1}^{t} + S_i^t \times \left(\vec{x}_{r2}^{t} - \vec{x}_{r3}^{t}\right) - \tau_3 \times \delta \times \gamma_t \times S_i^t\right) \tag{9}$$

$$w_i = \frac{\sum_{n=1}^{N} u_{i,n}^2}{\sum_{n=1}^{N} (u_n)^2} \tag{10}$$

where $E_{entropy}$ denotes the modal energy entropy normalized to [0, 1]; w_i is the energy weight of each mode; u_i represents the decomposed modal component; and k is the total number of modes.

(b) Modal Orthogonality

Orthogonality measures the independence and uncorrelated nature between modal components. Complete orthogonality among modes facilitates accurate signal interpretation and modeling. In contrast, a strong correlation between modes may indicate redundancy or signal leakage.

For any two modes u_i and u_j, the orthogonality index is calculated as follows:

$$o(i,j) = \frac{|\langle u_i, u_j \rangle|}{N \cdot \sigma(u_i) \cdot \sigma(u_j)} \tag{11}$$

The overall score of modal orthogonality is calculated as the average of all pairwise combinations:

$$O = \frac{1}{\binom{k}{2}} \sum_{1 \le i < j \le k} o(i,j) \tag{12}$$

where $o(i,j)$ denotes the orthogonality between modal components; O is the overall orthogonality index; $\langle u_i, u_j \rangle$ represents the dot product between u_i and u_j; and $\sigma(\cdot)$ is the standard deviation.

(c) Energy Concentration in the Dominant Frequency Band

The energy concentration in the dominant frequency band is used to measure the frequency-domain separability of the modal components. A dispersed distribution of dominant frequencies indicates that load characteristics have been effectively distinguished, enhancing the physical interpretability of the decomposition. In contrast, overlapping dominant frequencies may lead to modal redundancy and reduced decomposition quality.

For each mode, the dominant frequency is identified as the frequency corresponding to the maximum amplitude after applying the fast Fourier transform (FFT). Let the position of the dominant frequency band be denoted as:

$$f_i = argmax(FFT(u_i)) \tag{13}$$

$$U_p = \frac{Count\{round(f_i)\}_{i=1}^{k}}{k} \tag{14}$$

where $FFT(\cdot)$ denotes the fast Fourier transform; $Count(\cdot)$ represents the deduplication and counting operation; $round(\cdot)$ refers to the numerical discretization process; and U_p is the proportion of unique dominant frequencies, which means the energy concentration in the dominant frequency band.

(d) Alpha Penalty Factor

To prevent excessively high values of α during the optimization process—which may cause the overall decomposition structure to deviate from expectations—an Alpha penalty factor is introduced to constrain abnormal α values. This ensures a balanced trade-off between energy distribution and structural rationality in the decomposition results.

$$\alpha_p = \frac{\lambda_1}{1 + exp(-\lambda_2(minmax(\alpha) - \lambda_3))} \tag{15}$$

where α_p denotes the penalty factor; $minmax(\cdot)$ represents the normalization operation; and λ_i is the coefficient of the penalty term.

(e) Fitness Function

The final fitness function comprehensively considers multiple indicators, including modal energy entropy, orthogonality, dominant frequency distinguishability, and the Alpha penalty factor. Its formulation is given as follows:

$$F = \big(\gamma E_{entropy} + (1-\gamma)O\big) U_p \alpha_p \tag{16}$$

where γ is the weighting coefficient.

MCPO is selected within the framework to address the challenges of complexity and non-stationarity in processing user-side load data by integrating a multi-dimensional and physically interpretable fitness function. Unlike traditional optimizers that typically rely on

a single objective, the MCPO-based method evaluates decomposition quality using criteria such as modal energy entropy, orthogonality, and dominant frequency concentration. This customized evaluation approach ensures that the parameter optimization for VMD not only pursues mathematical performance but also aligns with the physical characteristics of the load data.

2.1.3. Frequency-Domain Feature Enhancement (FDFE)

To further extract the latent frequency-domain features embedded in intrinsic mode functions (IMFs), a frequency-domain feature enhancement (FDFE) module is designed in this study. First, each IMF component is transformed from the time domain to the frequency domain using fast Fourier transform (FFT), and a peak detection algorithm is applied to identify the dominant and significant secondary frequency components. Second, an adaptive weighting strategy is employed to selectively enhance key frequency bands, while a dynamic energy normalization mechanism is introduced to maintain the stability of the time-domain signal. Finally, the enhanced frequency-domain features are fused with the original IMF components and meteorological features to construct a multimodal augmented dataset, providing higher-quality input features for subsequent forecasting models.

(a) Each intrinsic mode function (IMF) is transformed from the time domain to the frequency domain using fast Fourier transform (FFT), and its amplitude spectrum is calculated as follows:

$$U(k) = \sum_{t=0}^{n-1} u(t) e^{-j2\pi kt/n},\ k = 0, 1, \ldots, n-1 \tag{17}$$

$$A(k) = |U(k)| = \sqrt{\mathrm{Re}\{U(k)\}^2 + \mathrm{Im}\{U(k)\}^2} \tag{18}$$

where $u(t)$ is the intrinsic mode function (IMF); $U(k)$ is the complex spectral value of the k-th frequency component corresponding to the IMF; $A(k)$ is the amplitude of the k-th frequency component; and $\mathrm{Re}\{\cdot\}$ and $\mathrm{Im}\{\cdot\}$ represent the real and imaginary parts, respectively.

(b) Significant Peak Detection

The average amplitude within the half-spectrum segment (from DC to the Nyquist frequency) is calculated as follows:

$$\mu_f = \frac{1}{\lfloor n/2 \rfloor} \sum_{k=0}^{\lfloor n/2 \rfloor - 1} A(k) \tag{19}$$

All peak indices satisfying the condition $A(k) \geq \mu_f$ are identified and stored in the set $\mathcal{P}$, and the dominant frequency index along with its corresponding amplitude is determined.

$$\mathcal{P} = \{Peak(k|A(k)) = True\ \&\ A(k) \geq \mu_f\} \tag{20}$$

$$k_m = \underset{k \in \mathcal{P}}{\mathrm{argmax}} A(k) \tag{21}$$

$$A_m = A(k_m) \tag{22}$$

where $Peak(\cdot)$ is the peak detection function, and $argmax(\cdot)$ is the maximum value selection function.

(c) Frequency-Domain Enhancement

Based on the detected dominant and significant secondary frequency components, an adaptive weighting strategy is employed to selectively enhance key frequency bands $U'(k)$:

$$U'(k) = \begin{cases} U(k)\Big(\ 1+u_1 \ \Big), & k = k_m \ or \ k = n - k_m \\ U(k)\Big(\ 1+u_2 \ \Big), & k \in \mathcal{P} \setminus \{k_m\} \ \& A(k) > \tau A_m \\ U(k), & else \end{cases} \tag{23}$$

To ensure the stability of the time-domain signal, dynamic energy normalization is applied to the enhanced signal:

$$\widetilde{u}(t) = \frac{1}{n} \sum_{k=0}^{n-1} U'(k) e^{j2\pi kt/n}, \ t = 0, 1, \ldots, n-1 \tag{24}$$

$$u'(t) = \widetilde{u}(t) \times \frac{\sigma_u}{\sigma_{\widetilde{u}}} \tag{25}$$

The enhanced signal is fused with the original intrinsic mode functions (IMFs) and meteorological features to generate the final frequency-domain augmented dataset:

$$X = [u_k, weather] \rightarrow X_{fusion} = [u_k, weather, \widetilde{u}(t)] \tag{26}$$

where u_1 is the dominant frequency enhancement coefficient; u_2 is the secondary frequency enhancement coefficient; τ is the significance threshold for secondary frequencies; $\widetilde{u}(t)$ is the time-domain signal after enhancement; σ_u and $\sigma_{\widetilde{u}}$ represent the standard deviations of the original intrinsic mode function and the enhanced signal, respectively; X and X_{fusion} denote the model input data before and after feature enhancement; and *weather* refers to the original meteorological features.

The originality of the proposed FDFE module does not lie in the FFT operation itself, but in embedding FFT within a task-oriented enhancement framework. Through adaptive thresholding and weighting strategies, FDFE selectively amplifies informative frequency components while suppressing noise and redundancy. This domain-specific adaptation enables the decomposition results to align more effectively with the non-stationary characteristics of user-side load data, thereby providing clearer and more representative features for downstream forecasting tasks.

2.2. MSCPatchTST (PatchTST with Multi-Scale Convolution)

MSCPatchTST uses patches of the time series as input units. By introducing a patching mechanism, the original time series is divided into multiple non-overlapping patches, which overcomes the limitations of traditional step-by-step processing. In addition, multi-scale convolution is incorporated to enhance the model's ability to capture local and multi-scale information. The architecture of MSCPatchTST is illustrated in Figure 3.

$$X_{1D}^{l} = AETFFBlock\left(X_{1D}^{l-1}\right) + X_{1D}^{l-1} \tag{27}$$

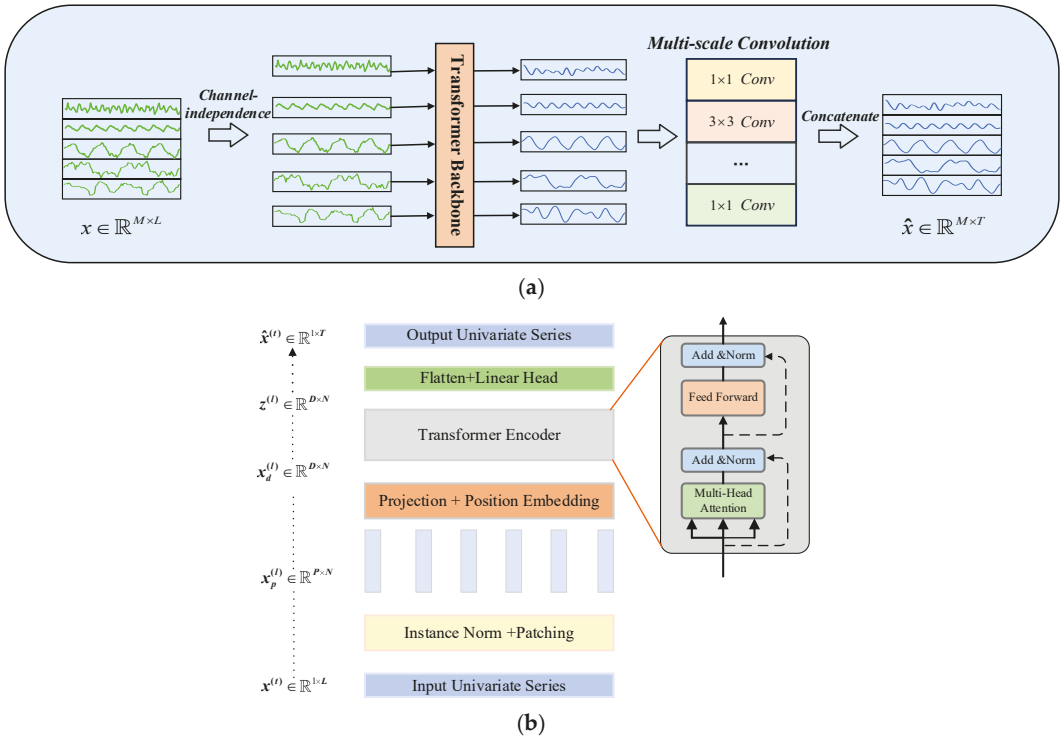


Figure 3. MSCPatchTST Framework. (**a**) Overall structure of MSCPatchTST. (**b**) Schematic diagram of transformer structure.

2.2.1. Channel Independence

PatchTST adopts channel-independent input for time series, where each variable is processed separately. Its core architecture is a Transformer encoder, which processes a given sequence of data $x_1, \ldots, x_L$ using a look-back window. The sequence $x_1, \ldots, x_L$ is decomposed into M univariate sequences $x(i) = [x1(i), \ldots, xL(i)] \in \mathbb{R}^{1 \times L}$, where $i = 1, \ldots, M$. These sequences are independently fed into the Transformer for processing. After computation, the Transformer outputs the final prediction results as follows:

$$\hat{\mathbf{x}}^{(i)} = \hat{\mathbf{x}}_{L+1}^{(i)}, \ldots, \hat{\mathbf{x}}_{L+T}^{(i)} \in \mathbb{R}^{1 \times T} \tag{28}$$

2.2.2. Patching

When processing a univariate time series, it is first divided into multiple patches. Let P be the patch length, and S be the stride, i.e., the non-overlapping interval between two consecutive patches. After patching, a patch sequence $\mathbf{x_p^{(i)}} \in \mathbb{R}^{\mathbf{P} \times \mathbf{N}}$ is generated, where N denotes the number of resulting patches. The calculation is given by the following formula: $\mathbf{N} = \left[\frac{\mathbf{L}-\mathbf{P}}{\mathbf{S}}\right] + 2$, where L is the length of the input sequence.

2.2.3. Multi-Scale Convolution Module

Although PatchTST is effective at capturing global information, its ability to extract local and multi-scale features remains limited. This limitation is particularly evident in time series data, where local details and global trends may vary across different temporal scales. Relying on a single-scale processing approach may result in the loss of valuable information.

To address the above limitations, an MSCBlock is introduced to extract multi-scale features and perform adaptive feature fusion. In addition, a residual connection is incorporated to ensure that the enhanced features retain the original information. The detailed mechanism of the MSCBlock is as follows:

(a) The input is processed using 1D convolutions with different kernel sizes. Suppose there are $K = [k_1, k_2, \ldots, k_K]$ convolutional kernels. For the i-th convolutional layer, the operation is defined as:

$$y_i = Conv1D_{(k_i)}(x_{reshape}) \tag{29}$$

(b) The multi-scale temporal features are concatenated along the convolutional channel dimension:

$$Y_{concat} = Concat(y_1, y_2, \ldots, y_K) \tag{30}$$

(c) A 1×1 convolution is applied in combination with a residual connection to adaptively fuse the multi-scale features.

$$Y_{fused} = ReLU\Big(Conv1D_{(1\times1)}(Y_{concat})\Big) \tag{31}$$

$$Y_{output} = x_{reshape} + Y_{fused} \tag{32}$$

where $Conv1D(\cdot)$ denotes the 1D convolution operation; y_i is the output of the i-th convolutional layer; Y_{concat} represents the concatenated multi-scale tensor; $Concat(\cdot)$ denotes the concatenation operation; $ReLU(\cdot)$ is the activation function; Y_{fused} is the fused multi-scale feature; and Y_{output} represents the final output feature tensor.

2.2.4. Transformer Encoder

In PatchTST, each patch is first projected into the Transformer hidden space through a trainable linear projection $W_p \in \mathbb{R}^{\mathcal{D}\times\mathcal{P}}$ and a learnable positional encoding $W_{\text{pos}} \in \mathbb{R}^{\mathcal{D}\times N}$, enabling better capture of temporal features within patches. Then, a multi-head attention mechanism is applied to transform the processed data into the query (Q), key (K), and value (V) matrices. Finally, a fully connected layer with a linear head is used to generate the final prediction results:

$$\hat{x}^{(i)} = \left[\hat{x}^{(i)}_{L+1}, \ldots, \hat{x}^{(i)}_{L+T}\right] \in \mathbb{R}^{1\times T} \tag{33}$$

3. Examples Analysis

3.1. User-Side Load Data

The experimental data are derived from a publicly available user-side load dataset from a province in China. The dataset includes separate load records for weekdays, Saturdays, and Sundays. The selected time period spans from 2 January 2024 to 17 March 2025, with a sampling interval of 30 min, resulting in a total of 12,576 data points. To eliminate the influence of dimensional inconsistencies among different variables, all input data for the neural network were normalized to the range $[-1, 1]$. Inverse normalization was then applied to the predicted values in the test set based on the statistical characteristics of the training set. The dataset was split into training, validation, and test sets in a ratio of 6:2:2.

3.2. Model Parameter Settings

The proposed model is implemented in a Python 3.11 environment using the PyTorch 2.3 framework, and all experiments are conducted on a system equipped with a Core™ i5-13500HX CPU (2.50 GHz) and 8 GB of RAM. Given the lack of established theoretical guidelines for hyperparameter selection, the initial parameter settings are determined empirically. These parameters are further fine-tuned based on model performance during subsequent experiments. The final hyperparameters, determined after multiple trials, are summarized in Table 1.

Table 1. Parameter Settings.

Parameter Name	Parameter Value
Seq len	144
Number of attention heads	8
Epoch	100
Batch size	24
Patience	10
Dropout	0.05
Optimizer	Adam
Learning rate	0.001
Activation function	GELU
Loss function	MSE

3.3. Evaluation Metrics

In this study, the model performance is evaluated using three metrics: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). Lower values of MAE, RMSE, and MAPE indicate smaller deviations between the predicted and actual values, reflecting higher prediction accuracy. The corresponding formulations are given in Equations (34)–(36):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i'| \tag{34}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{35}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - y_i'|}{y_i}\cdot 100\% \tag{36}$$

where n denotes the number of data points, y_i represents the actual values, y'_i denotes the predicted values, and $\overline{y_i}$ is the mean of the actual values.

3.4. MCPO-VMD Decomposition Results

VMD can decompose a non-stationary signal into K components within limited bandwidths. However, its performance is highly sensitive to the selection of the parameters K and α. An excessively large K may lead to mode mixing, while a small K may result in insufficient decomposition. Similarly, the value of α also affects the decomposition quality. To address this issue, the MCPO algorithm is employed to optimize the VMD process, enabling precise determination of the optimal K–α parameter combination and thereby enhancing the decomposition performance. The optimized VMD can transform a complex signal into multiple simpler components, reduce data complexity, reveal multi-frequency characteristics, improve input data quality, and ultimately enhance the model's signal processing and forecasting accuracy.

Taking the weekday dataset as an example, the specific decomposition results of VMD optimized by MCPO are shown in Figure 4.

From Figure 4, the optimized VMD uses MCPO to determine the optimal K–α. It effectively decomposes into 9 IMF components. IMF 1 has a relatively high frequency. It reflects high-frequency signal fluctuations. IMF 9 has a low frequency. It shows the basic trend. Each component has limited bandwidth. There is no obvious mode mixing. This verifies that parameter optimization improves decomposition quality. It provides clearer multi-frequency features for subsequent model input.

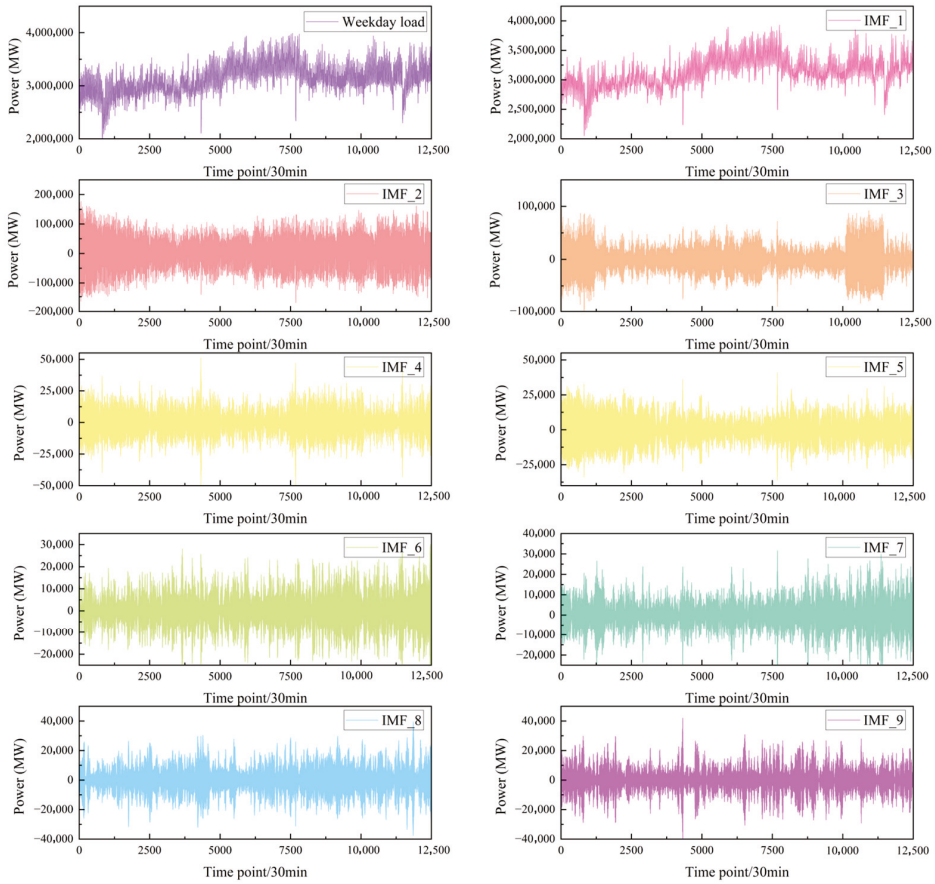


Figure 4. VMD Decomposition Results.

3.5. Ablation Experiment

The results of the ablation study are presented in Table 2 and Figures 5–7. The experimental findings demonstrate that the proposed MCPO-VMD module, FDFE module, and MSC module each contribute to improved forecasting performance on user-side load data across weekdays, Saturdays, and Sundays during the period from 2 January 2024 to 17 March 2025.

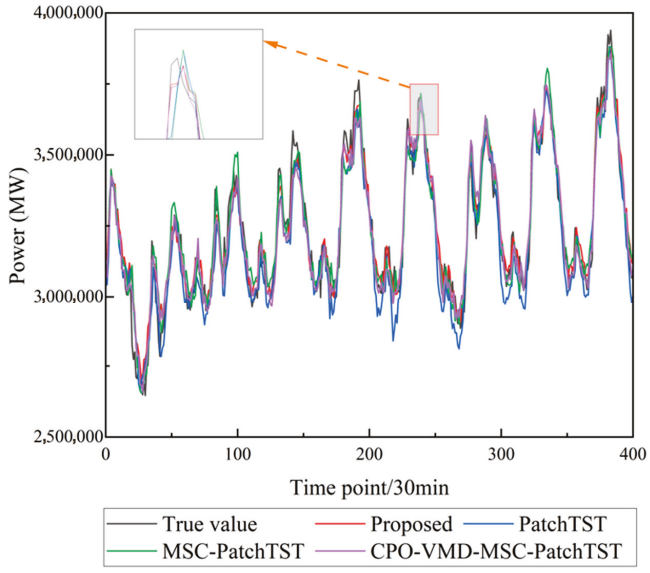


Figure 5. Ablation Study on Weekdays Load Data.

Table 2. Ablation Experiment Prediction Metrics on User-Side Load Data.

Data Type	Method	RMSE	MAE	MAPE%
Weekdays	PatchTST	89,929.68	69,955.73	2.1973
	MSCPatchTST	76,526.31	58,820.48	1.8664
	CPO-VMD-MSCPatchTST	68,999.37	50,619.42	1.6174
	MCPO-VMD-MSCPatchTST	67,438.94	50,379.3	1.6001
	Proposed	59,131.17	44,954.44	1.4384
Saturdays	PatchTST	158,580.61	122,474.26	3.926
	MSCPatchTST	145,094.88	112,955.72	3.6334
	CPO-VMD-MSCPatchTST	127,526.27	96,752.75	3.1212
	MCPO-VMD-MSCPatchTST	126,446.16	95,918.71	3.0941
	Proposed	123,069.89	93,108.72	3.0026
Sundays	PatchTST	136,026.05	107,074.64	3.6329
	MSCPatchTST	111,315.49	86,603.41	2.9148
	CPO-VMD-MSCPatchTST	110,783.8	89,448.47	3.0025
	MCPO-VMD-MSCPatchTST	90,260.43	68,000.33	2.2815
	Proposed	86,655.03	64,351.65	2.2084

Specifically, after incorporating the MSC module, the model is able to extract multi-scale features and adaptively fuse them, resulting in average reductions of 13.25%, 13.76%, and 13.77% in RMSE, MAE, and MAPE, respectively. This confirms the module's effectiveness in capturing local and multi-scale information. In addition, the integration of the MCPO-VMD module allows the model to optimize the VMD process via the MCPO algorithm, significantly reducing data complexity and extracting multi-frequency characteristics. This improves the quality of the model input and achieves average reductions of 26.05%, 28.40%, and 28.52% in RMSE, MAE, and MAPE, respectively, demonstrating its enhanced ability to process non-stationary signals. Furthermore, with the addition of the FDFE module, the model can extract dominant and significant secondary frequency components in the frequency domain and enhance their key responses through an adaptive weighting mechanism. This leads to average improvements of 30.32%, 32.37%, and 31.84% in RMSE, MAE, and MAPE, respectively, validating the module's capability in enhancing the perception of frequency-domain structural features. Overall, the proposed multi-dimensional CPO-VMD-MSC-PatchTST model, with the joint optimization of the CPO-VMD and MSC modules, significantly outperforms the baseline PatchTST. Compared to PatchTST, it achieves average reductions of 30.32%, 32.37%, and 31.84% in RMSE, MAE, and MAPE, respectively, confirming the effectiveness of the proposed modules in improving forecasting accuracy.

To validate the effectiveness of key components individually, ablation studies were conducted. First, to assess the optimization role of the customized fitness function in MCPO, "CPO-VMD-MSCPatchTST" was compared with "MCPO-VMD-MSCPatchTST." The results show that the weekday RMSE decreased from 68,999.37 to 67,438.94, and on Sunday, the RMSE dropped from 110,783.8 to 90,260.43, with the MAPE% decreasing from 3.0025 to 2.2815. This indicates that the MCPO-based fitness function improves the accuracy of VMD parameter selection, thereby reducing prediction error.

Furthermore, to verify the value of the multi-scale convolution module, the baseline "PatchTST" model was compared with "MSCPatchTST," which incorporates the module. Experimental data showed that the weekday RMSE decreased from 89,929.68 to 76,526.31, the Saturday MAPE% decreased from 3.926 to 3.6334, and the Sunday RMSE decreased from 136,026.05 to 111,315.49. This result demonstrates that the introduction of multi-size convolution kernels enhanced the model's capability for extracting multi-scale load features.

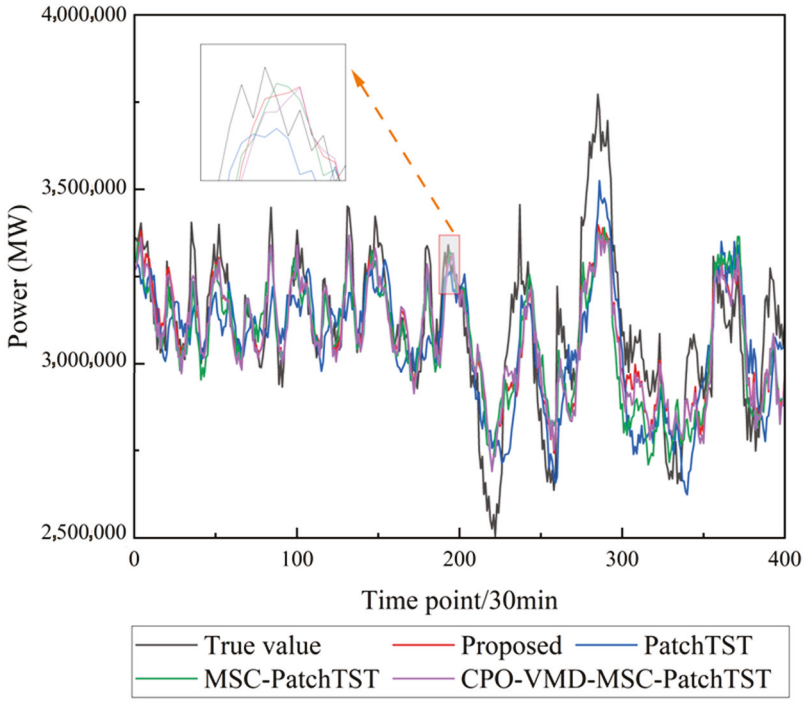


Figure 6. Ablation Study on Saturdays Load Data.

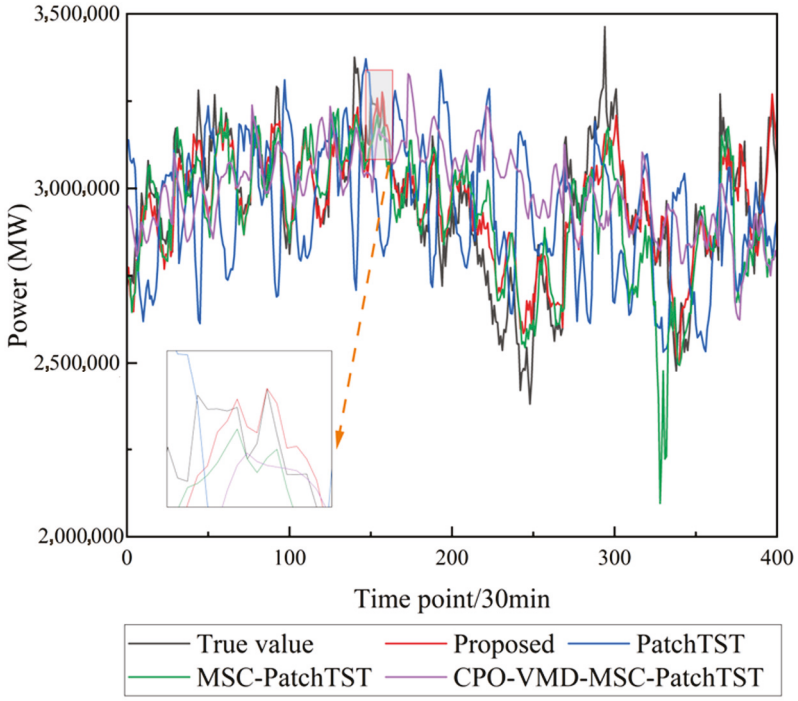


Figure 7. Ablation Study on Sundays Load Data.

From Figure 5, the proposed model is closer to the true value throughout the prediction period, especially in the high-fluctuation regions. Compared with PatchTST, MSC-PatchTST, and CPO-VMD-MSC-PatchTST, it shows smaller deviations, which is consistent with the lower error metrics in Table 2. Figure 6 presents a similar pattern. The proposed model can better capture the load variation characteristics on Saturdays. In the local enlarged area, its prediction curve fits the true value more closely, effectively reducing the overestimation or underestimation phenomena that occur in other comparative models. For Figure 7, the proposed model also demonstrates excellent prediction performance. Even in the regions

with significant load fluctuations, it can follow the trend of the true load values well, further verifying the effectiveness of the integrated modules in improving the model's adaptability to different load patterns on Sundays.

3.6. Comparative Experiment

To evaluate the effectiveness of the proposed method in user-side load forecasting, several baseline models—including Autoformer, Transformer, FEDformer, iTransformer, and PatchTST—are selected for comparison. The forecasting performance of all models on user-side load data is summarized in Table 3 and Figures 8–10. Experimental results show that the proposed MCPO-VMD-FDFE-MSCPatchTST model consistently outperforms all baseline models across the three datasets (weekdays, Saturdays, and Sundays). Compared to the baseline models, the proposed model achieves an overall RMSE reduction of 45.65% across the three user-side load forecasting tasks. Taking the weekday dataset as an example, the MCPO-VMD-FDFE-MSCPatchTST model achieves average reductions of 51.01%, 51.69%, and 51.24% in RMSE, MAE, and MAPE, respectively, compared to the other models. These results confirm the proposed model's significant advantages and practical effectiveness in modeling non-stationary load sequences and improving forecasting accuracy.

Table 3. Comparative Experiment 1: Prediction Metrics on User-Side Load Data.

Data Type	Method	RMSE	MAE	MAPE%
Weekdays	Autoformer	202,945.7	157,396.15	4.9929
	Transformer	188,512.8	150,230.16	4.7268
	FEDformer	221,169.5	172,782.76	5.372
	iTransformer	78,522.16	59,294.69	1.8536
	PatchTST	89,929.68	69,955.73	2.1973
	Proposed	76,526.31	58,820.48	1.8664
Saturdays	Autoformer	246,141.07	195,330.32	6.2523
	Transformer	238,694.39	180,513.21	5.9675
	FEDformer	228,628.75	181,450.29	5.9254
	iTransformer	204,594.55	144,689.18	4.7746
	PatchTST	158,580.61	122,474.26	3.926
	Proposed	145,094.88	112,955.72	3.6334
Sundays	Autoformer	272,625.66	222,147.36	7.6234
	Transformer	219,467.04	177,725.57	6.0307
	FEDformer	208,630.76	159,150.79	5.5775
	iTransformer	178,165.15	130,458.19	4.4611
	PatchTST	136,026.05	107,074.64	3.6329
	Proposed	111,315.49	86,603.41	2.9148

The visualization in Figure 8 shows that the proposed model's prediction curve has a higher degree of agreement with the true values over the entire forecast period, particularly in high-fluctuation intervals such as peak and valley loads. Compared to baseline models like Autoformer and Transformer, this model achieves a better fit, which is consistent with the quantitative error metrics in Table 3. This result indicates that the model is capable of capturing the high-frequency, multi-peak-valley fluctuation characteristics of weekday loads. From Figure 9, it can be seen that the proposed model's prediction curve is able to follow the trend of the true values. At the locally magnified load mutation points, this model's prediction deviation is smaller than that of other compared models. This suggests that the model adapts well to the unique load fluctuation patterns of Saturdays, thereby reducing overestimation or underestimation in the prediction results. As shown in Figure 10, when faced with scenarios of sharp drops or surges in Sunday loads, the proposed model's predictions remain largely consistent with the trend of the true values.

This reflects the model's stable performance in handling complex, non-stationary Sunday load sequences and demonstrates its generalization ability across different scenarios.

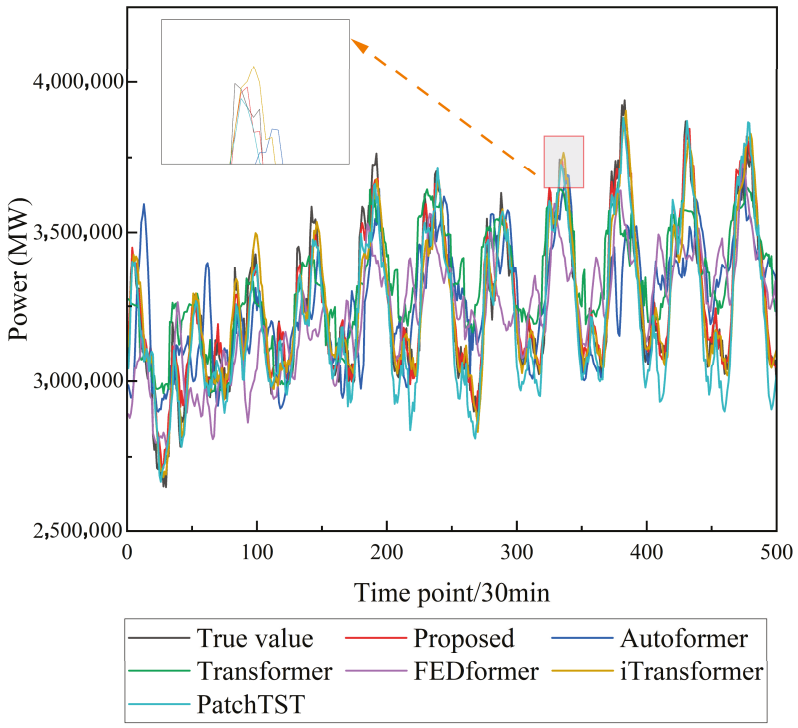


Figure 8. Comparison Experiment a on Weekdays Load Data.

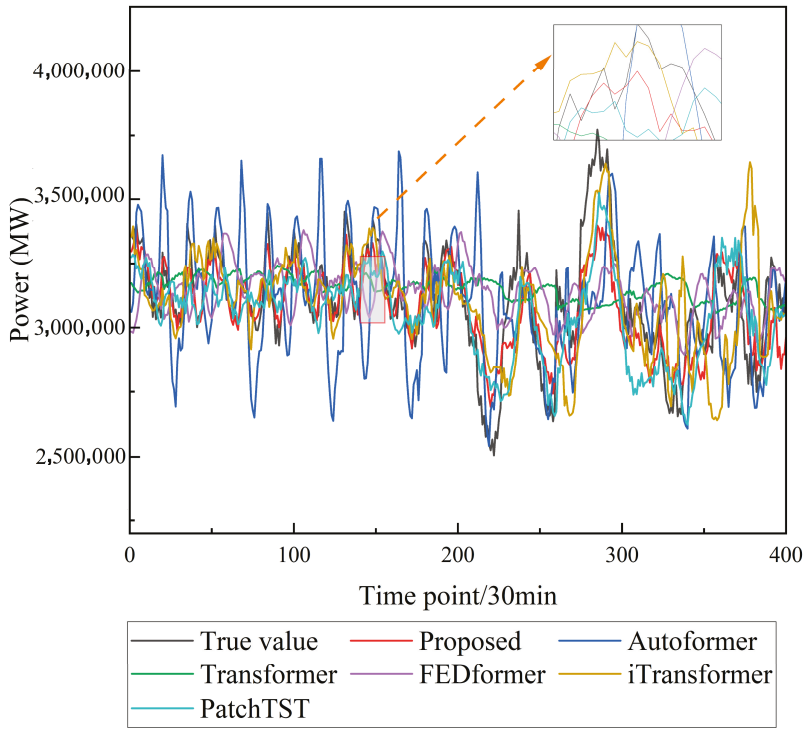


Figure 9. Comparison Experiment a on Saturdays Load Data.

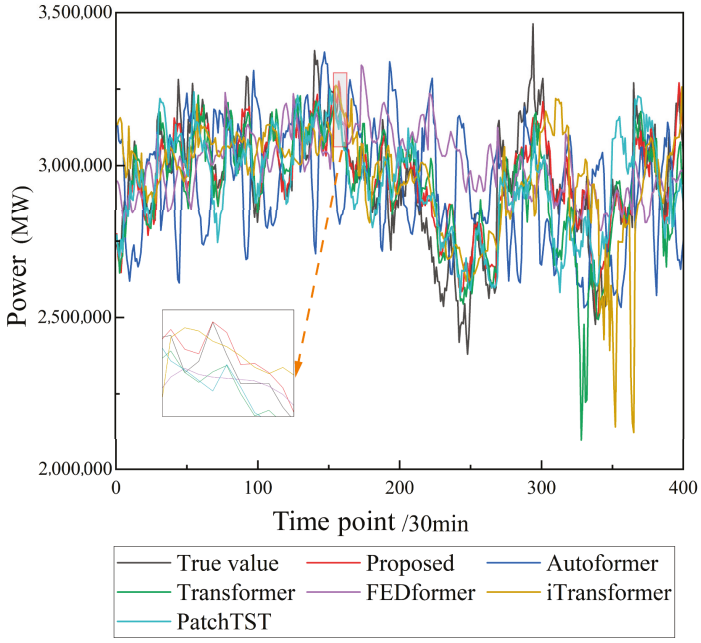


Figure 10. Comparison Experiment a on Sundays Load Data.

To further verify the effectiveness of the proposed integrated modeling strategy, the MCPO-VMD-FDFE decomposition-enhancement framework is embedded into five backbone models—Autoformer, Transformer, FEDformer, iTransformer, and PatchTST—for comparative experiments. The forecasting performance of each hybrid model on user-side load prediction tasks is presented in Table 4 and Figures 11–13. The experimental results show that the proposed MCPO-VMD-FDFE-MSCPatchTST model consistently achieves superior performance across all three user-side load datasets: weekdays, Saturdays, and Sundays. Compared to the other hybrid models, it achieves an overall reduction of 46.25% in RMSE across the three datasets. Taking the weekday dataset as an example, the MCPO-VMD-FDFE-MSCPatchTST model outperforms all other models with average reductions of 49.98%, 50.68%, and 50.45% in RMSE, MAE, and MAPE, respectively. These results demonstrate the strong effectiveness and robustness of the proposed model in user-side load forecasting tasks.

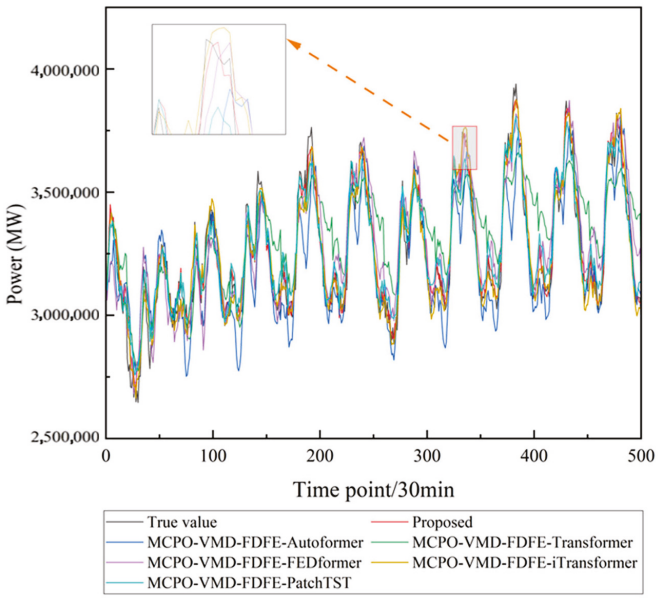


Figure 11. Comparison Experiment b on Weekdays Load Data.

Table 4. Comparative Experiment 2: Prediction Metrics on User-Side Load Data.

Data Type	Method	RMSE	MAE	MAPE%
Weekdays	MCPO-VMD-FDFE-Autoformer	155,956.6	116,683.01	3.7178
	MCPO-VMD-FDFE-Transformer	156,730.6	127,942.28	4.0373
	MCPO-VMD-FDFE-FEDformer	140,537.6	103,701.17	3.3195
	MCPO-VMD-FDFE-iTransformer	65,562.17	50,632.98	1.5981
	MCPO-VMD-FDFE-PatchTST	74,270.74	57,654.24	1.8427
	Proposed	59,131.17	44,954.44	1.4384
Saturdays	MCPO-VMD-FDFE-Autoformer	239,407.91	186,681.7	6.1231
	MCPO-VMD-FDFE-Transformer	220,768.05	170,794.68	5.5672
	MCPO-VMD-FDFE-FEDformer	204,852.45	152,448.84	5.0609
	MCPO-VMD-FDFE-iTransformer	184,007.49	133,593.04	4.4049
	MCPO-VMD-FDFE-PatchTST	136,738.68	103,215.95	3.3602
	Proposed	123,069.89	93,108.72	3.0026
Sundays	MCPO-VMD-FDFE-Autoformer	243,951.2	186,081.62	6.4431
	MCPO-VMD-FDFE-Transformer	209,132.61	171,928.29	5.7637
	MCPO-VMD-FDFE-FEDformer	200,642.97	149,461.14	5.2794
	MCPO-VMD-FDFE-iTransformer	139,269.59	110,671.9	3.8287
	MCPO-VMD-FDFE-PatchTST	91,299.71	74,078.26	2.5305
	Proposed	86,655.03	64,351.65	2.2084

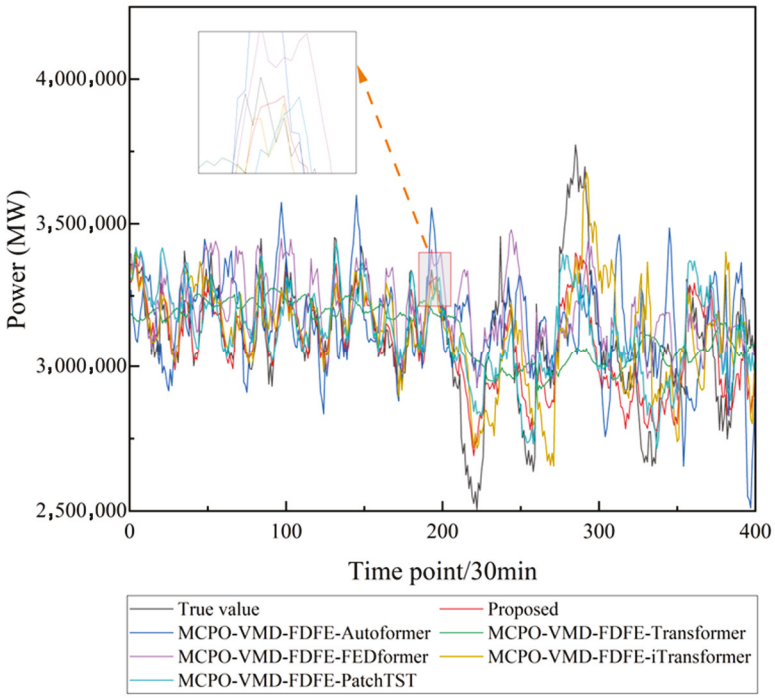


Figure 12. Comparison Experiment b on Saturdays Load Data.

From Figure 11, the proposed model aligns much closer with the true values throughout the entire prediction period, especially in high-fluctuation intervals with frequent load peaks and valleys. Its fitting degree surpasses that of hybrid models like MCPO-VMD-FDFE-Autoformer and MCPO-VMD-FDFE-Transformer. This clearly demonstrates the model's ability to accurately fit the weekday load patterns and verifies its effectiveness in capturing high-frequency, multi-peak-valley fluctuation characteristics. From Figure 12, the prediction curve of the proposed model closely follows the changes in true values. In locally enlarged regions with load mutation points, it shows smaller deviations compared to other hybrid models. This strongly proves that the model can precisely adapt to the unique fluctuation patterns of Saturday loads arising from differences in work-rest schedules and electricity consumption habits, reducing over-prediction or under-prediction situations.

From Figure 13, when facing extreme fluctuations of Sunday loads with scenarios of significant drops or sudden surges, the proposed model can still stably align with the trend of true values. This highlights the advantages of the integrated modules in handling complex, non-stationary Sunday load sequences and further supports the model's generalization ability across multiple scenarios.

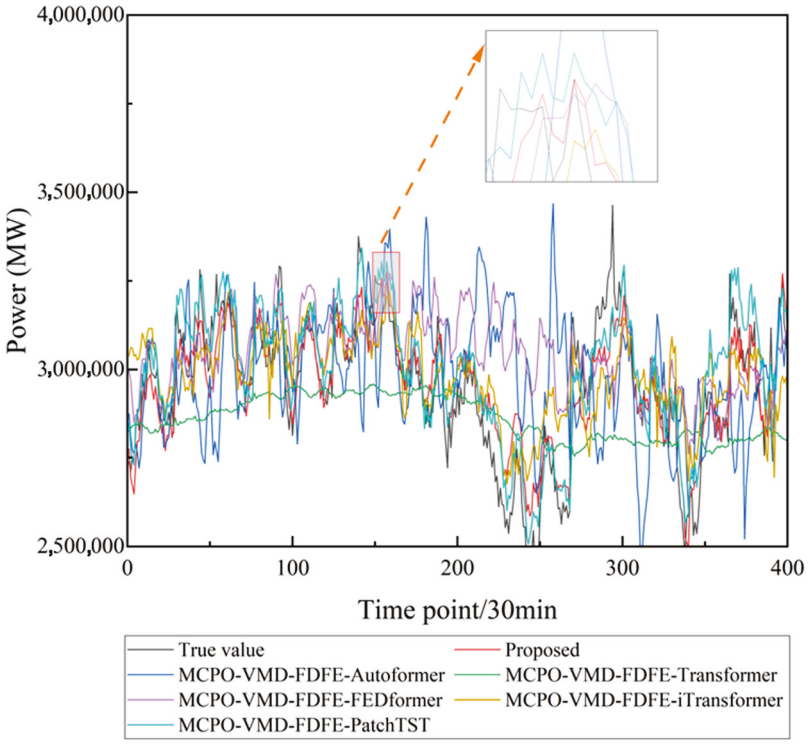


Figure 13. Comparison Experiment b on Sundays Load Data.

4. Conclusions

To address the limitations of poor adaptability in modal decomposition and insufficient frequency-domain feature representation in short-term user-side load forecasting tasks, this study proposes a novel forecasting method based on the MCPO-VMD-FDFE-MSCPatchTST framework. The effectiveness of the proposed method is verified through case studies, and the following conclusions are drawn:

(1) This study proposes a parameter optimization method based on MCPO with a multi-dimensional fitness function, which overcomes the limitations of traditional empirical parameter selection to better match the multi-frequency characteristics of user-side load data. This method enables more effective optimization of key VMD parameters, and ablation studies show that its inclusion leads to average reductions of 26.05%, 28.40%, and 28.52% in RMSE, MAE, and MAPE, respectively, demonstrating its capability in handling non-stationary signals.

(2) A frequency-domain feature enhancement (FDFE) module is designed to highlight key frequency components through adaptive thresholds and weighting strategies. This significantly improves the representational capacity of modal components. Ablation study results indicate that the inclusion of this module leads to average reductions of 30.32%, 32.37%, and 31.84% in RMSE, MAE, and MAPE, respectively, confirming its effectiveness in enhancing the model's perception of frequency-domain structural features.

(3) The integration of a multi-scale convolution module into the PatchTST model enables the collaborative extraction and adaptive fusion of both local and multi-scale features.

This enhances the model's ability to capture diverse patterns in user-side load data. Ablation study results show that this module contributes to average reductions of 26.05%, 28.40%, and 28.52% in RMSE, MAE, and MAPE, respectively.

(4) The proposed method organically combines the MCPO-VMD decomposition, FDFE, and MSCPatchTST modules into a unified framework. This integrated approach achieves favorable forecasting performance across all three user-side load datasets (weekdays, Saturdays, and Sundays). Experimental results demonstrate that the proposed method consistently outperforms all comparison models, with an overall RMSE reduction of 45.65%, fully validating its effectiveness and stability in user-side load forecasting tasks.

(5) Under the dataset and parameter settings used in this study, the complete workflow from training to prediction can be accomplished within approximately half an hour on a standard computing platform. This indicates that the proposed framework not only achieves high forecasting accuracy but also demonstrates satisfactory computational efficiency.

Author Contributions: Formal analysis and methodology and Writing—original draft and Supervision, Y.D.; Resources and Software and Data curation and Writing—original draft, J.S.; Formal analysis and methodology and Supervision, X.D.; Resources and Data curation and Writing—original draft, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to the proprietary nature of the dataset, which contains confidential information from the enterprise and cannot be publicly disclosed.

Conflicts of Interest: Author Yu Du was employed by the company NARI Nanjing Control System. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, W.; Chen, Y.; Xiao, C.; Yang, Y.; Yao, J. Design of short-term load forecasting method considering user behavior. *Electr. Power Syst. Res.* **2024**, *234*, 110529. [CrossRef]
2. Zhao, W.; Mu, G.; Zhu, Y.; Xu, L.; Zhang, D.; Huang, H. Research on electric load forecasting and user benefit maximization under demand-side response. *Int. J. Swarm Intell. Res.* **2023**, *14*, 1–20. [CrossRef]
3. Liu, Y.B.; Wu, H.; Liu, T.J.; Yang, Z.Y.; Liu, J.Y.; Li, Q.H. User-Side Net Load Forecasting Algorithm Integrating Empirical Mode Decomposition and Deep Learning. Autom. *Electr. Power Syst.* **2021**, *45*, 57–64.
4. Yan, Q.; Lu, Z.; Liu, H.; He, X.; Zhang, X.; Guo, J. An improved feature-time Transformer encoder-Bi-LSTM for short-term forecasting of user-level integrated energy loads. *Energy Build.* **2023**, *297*, 113396. [CrossRef]
5. Wei, L.; Zhao, J. Hybrid CNN-Multivariate LSTM Model for Accurate Short-Term Electricity Price Forecasting and Energy System Optimization. *J. Circuit Syst. Comput.* **2025**, *34*, 2550198. [CrossRef]
6. Yuqi, J.; An, A.; Lu, Z.; Ping, H.; Xiaomei, L. Short-term load forecasting based on temporal importance analysis and feature extraction. *Electr. Power Syst. Res.* **2025**, *244*, 111551. [CrossRef]
7. Li, Y.; Wang, H.; Huang, X.; Hao, J.; Lei, W.; Wang, Q. Short-term power load forecasting in distribution networks considering human comfort level. *Front. Energy Res.* **2025**, *13*, 1514755. [CrossRef]
8. Ahmad, A.; Xiao, X.; Mo, H.; Dong, D. TFTformer: A novel transformer based model for short-term load forecasting. *Int. J. Electr. Power Energy Syst.* **2025**, *166*, 110549. [CrossRef]
9. Liu, W.; Li, J. Short-term power load forecasting based on genetic algorithm improved VMD-BP. *Int. J. Intell. Inf. Technol.* **2025**, *21*, 1–18. [CrossRef]
10. Zhang, W.; Zhong, A.; Duan, K.; Shao, L. Ultra-short-term user-level load forecasting based on multi-scale component feature learning. *Power Syst. Technol.* **2024**, *48*, 2584–2592.
11. Cao, H.Z.; Wei, B.; Gao, C.; Wu, J.K.; Lei, Z.; Sui, Y.; Chen, Y.B. Short-Term Electric Load Forecasting Method Based on Deep Belief Network. *New Type Power Syst.* **2025**, *3*, 111–124.

12. Chen, C.; Ma, H.R.; Chen, L.J.; Ren, B.W.; Jin, C.; Zhang, T.Y. Multi-Step Short-Term Load Forecasting Method Based on User Group Segmentation. *High Volt. Eng.* **2023**, *49*, 4213–4222.
13. Moudgil, V.; Sadiq, R.; Brar, J.; Hewage, K. Dual-channel encoded bidirectional LSTM for multi-building short-term load forecasting. *J. Clean. Prod.* **2025**, *486*, 144555. [CrossRef]
14. Liu, F.; Wang, X.; Zhao, T.; Zhang, L.; Jiang, M.; Zhang, F. Novel short-term low-voltage load forecasting method based on residual stacking frequency attention network. *Electr. Power Syst. Res.* **2024**, *233*, 110534. [CrossRef]
15. Zhang, Q.; Chen, J.; Xiao, G.; He, S.; Deng, K. TransformGraph: A novel short-term electricity net load forecasting model. *Energy Rep.* **2023**, *9*, 2705–2717. [CrossRef]
16. Yazici, I.; Beyca, O.F.; Delen, D. Deep-learning-based short-term electricity load forecasting: A real case application. *Eng. Appl. Artif. Intell.* **2022**, *109*, 104645. [CrossRef]
17. Zhuang, H.; Guo, X.; Shan, D.; Tian, X.; Hou, T. Short-term power load forecasting model based on optimized VMD-BILSTM. In Proceedings of the 2023 8th Asia Conference on Power and Electrical Engineering (ACPEE), Tianjin, China, 14–16 April 2023; pp. 2029–2033. [CrossRef]
18. Zhang, Z.; Hong, W.C. Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. *Knowl. Based Syst.* **2021**, *228*, 107297. [CrossRef]
19. Yang, W.; Shi, J.; Li, S.; Song, Z.; Zhang, Z.; Chen, Z. A combined deep learning load forecasting model of single household resident user considering multi-time scale electricity consumption behavior. *Appl. Energy* **2022**, *307*, 118197. [CrossRef]
20. Tang, Q.F.; Liu, N.; Zhang, J.H.; Yu, Z.Z.; Zhang, Q.X.; Lei, J.Y. Short-Term Load Forecasting Method for User-Side Microgrids Based on EMD-KELM-EKF and Parameter Optimization. *Power Syst. Technol.* **2014**, *38*, 2691–2699.
21. Zhou, M.; Yu, J.; Wang, M.; Quan, W.; Bian, C. Research on the combined forecasting model of cooling load based on IVMD-WOA-LSSVM. *Energy Build.* **2024**, *317*, 114339. [CrossRef]
22. Li, H.; Tang, Y.; Liu, D. A CrossInformer model based on dual-layer decomposition and interpretability for short-term electricity load forecasting. *Alex. Eng. J.* **2025**, *129*, 117–127. [CrossRef]
23. Cui, X.; Zhang, X.; Niu, D. A new framework for ultra-short-term electricity load forecasting model using IVMD–SGMD two–layer decomposition and INGO–BiLSTM–TPA–TCN. *Appl. Soft Comput.* **2024**, *167*, 112311. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Interpretable Deep Prototype-Based Neural Networks: Can a 1 Look like a 0?

Esteban García-Cuesta *, Daniel Manrique and Radu Constantin Ionescu

Departamento de Inteligencia Artificial, ETSIINF, Universidad Politécnica de Madrid, 28040 Madrid, Spain; daniel.manrique@upm.es (D.M.); radu-constantin.ionescu@alumnos.upm.es (R.C.I.)
* Correspondence: esteban.garcia@upm.es; Tel.: +34-91-067-2887

Abstract: Prototype-Based Networks (PBNs) are inherently interpretable architectures that facilitate understanding of model outputs by analyzing the activation of specific neurons—referred to as prototypes—during the forward pass. The learned prototypes serve as transformations of the input space into a latent representation that more effectively encapsulates the main characteristics shared across data samples, thereby enhancing classification performance. Crucially, these prototypes can be decoded and projected back into the original input space, providing direct interpretability of the features learned by the network. While this characteristic marks a meaningful advancement toward the realization of fully interpretable artificial intelligence systems, our findings reveal that prototype representations can be deliberately or inadvertently manipulated without compromising the superficial appearance of explainability. In this study, we conduct a series of empirical investigations that demonstrate this phenomenon, framing it as a structural paradox potentially intrinsic to the architecture or its design, which may represent a significant robustness challenge for explainable AI methodologies.

Keywords: prototype-based networks; interpretable AI; robustness of explanations; neural networks

1. Introduction

Model interpretability is a critical requirement in the development of machine learning models intended to represent aspects of real-world phenomena. Numerous techniques and studies have sought to enhance interpretability, primarily through post hoc analyses or by employing surrogate models that are inherently more interpretable. Despite these efforts, a fundamental challenge persists: establishing a clear correspondence between the knowledge acquired by the model and observable elements of reality [1]. In an effort to address this issue, a Prototype Deep Neural Network (PDNN) was introduced in [2]. Prototype learning is a subclass of intrinsically interpretable methods, derived from the classical paradigm of case-based reasoning known as prototype classification. In this context, a prototype is not constrained to a single observation from the training set, but rather may constitute a generalized representation formed from a combination of multiple samples in the learned latent space [3]. The prototype learning framework typically incorporates an autoencoder in conjunction with a prototype classifier layer. The autoencoder generates a compressed latent representation of the input data, while the prototype layer identifies representative points within this space that both resemble specific inputs and are indicative of particular classes. Each learned prototype can subsequently be visualized through the decoder, enabling interpretability of the learned features, and can be traced back to the output

softmax layer, thereby providing insight into class associations. Moreover, this architectural design supports a form of case-based reasoning by identifying the training sample(s) most similar to a given prototype, further enhancing the transparency and accountability of the model's decision-making process.

In this context, while the interpretability of the method has received considerable attention, the reliability and validity of the learned prototypes have often been overlooked. For relatively simple datasets, such as handwritten digits [4], prototype validation is typically performed by reconstructing and visualizing the prototypes in the input space, followed by a qualitative assessment of whether the prototypes visually resemble the intended classes. However, this raises a critical question: can we confidently assert that a prototype which appears to represent, for instance, the digit "1", is indeed interpreted as such by the Deep Neural Network (DNN)? In this study, we address this question through a series of experiments, including the development of a novel Deep Prototype-Based Network. Our results demonstrate that visual similarity does not necessarily align with the model's internal representation, highlighting a potential gap between human interpretability and model-driven classification.

2. State of the Art

Deep networks can reason through learned prototypes while remaining end-to-end trainable [2]. Prototype-based neural networks offer a hybrid approach that combines the power of deep learning with the interpretability of case-based reasoning. Its architecture integrates a convolutional autoencoder with a prototype-based classification layer, allowing the model to explain its predictions through comparisons with prototypical training examples. Building upon this foundational work, ProtoPNet was proposed in [5] , that model popularized the "this-looks-like-that" paradigm through patch-to-prototype visualizations, which has also been extended in Semi-ProtoPNet [6] and Ps-ProtoPNet [7] to include rejection capabilities and the combination of models in Comb-ProtoPNet [8]. The training methodology employed techniques such as prototype projection and class-specific weight regularization, achieving competitive performance across fine-grained classification tasks involving bird, car, and dermatological image datasets, while preserving interpretable, image-level justifications.

Two primary strategies have been proposed to overcome the representational limitations inherent in single-layer prototype-based models. First, ProtoTree was introduced in [9], a differentiable decision tree architecture in which the internal nodes correspond to learned prototypes. This structure enables an input to traverse only $O(logK)$ nodes, producing compact and interpretable rule chains. Second, explicit prototype hierarchies [10] organize low-level visual parts into increasingly abstract semantic concepts, facilitating multi-granular reasoning and enhancing robustness to out-of-distribution inputs.

Hybrid approaches have also been developed to enable reasoning beyond raw pixel representations. Concept Bottleneck Models (CBMs) [11] structure the prediction pipeline around human-interpretable attributes, allowing users to intervene and correct misclassified concepts at inference time. In a complementary direction, the Deep k-Nearest Neighbours (DkNN) algorithm [12] enhances arbitrary classifiers by incorporating layer-wise nearest neighbor retrieval, thereby quantifying model uncertainty through non-conformity scores. Both methods underscore the utility of leveraging intermediate representations, particularly similarity structures within hidden layers, without compromising predictive accuracy, a principle that underpins the multilayer prototype architecture proposed in this work. Additionally, Capsule Networks [13] implement dynamic routing mechanisms to capture part-to-whole relationships, while part-based R-CNNs [14] localize discriminative object regions.

The Deep Prototype Network (DPN) introduced in this work adopts a compositional reasoning framework akin to prior hierarchical or part-based models, but implements this structure through a deliberately layered architecture rather than relying on post hoc clustering of prototypes. This design choice is motivated by the principle that successive prototype layers should progressively capture increasingly abstract semantic concepts or larger spatial extents, thereby enabling richer part-to-whole reasoning directly within the network's forward pass.

3. Deep Prototype-Based Network Architecture

Figure 1 illustrates the proposed Deep Prototype-Based Network (DPBN) architecture, which extends [2] notation, cost terms, and design philosophy, but stacks several prototype layers in a feedforward chain rather than using a single one. Therefore, each layer has similar architectural complexity to [2] (including the convolutional neural network). The proposed DPBN performs a back-and-forth data flow that couples representational capacity with prototype-based interpretability.

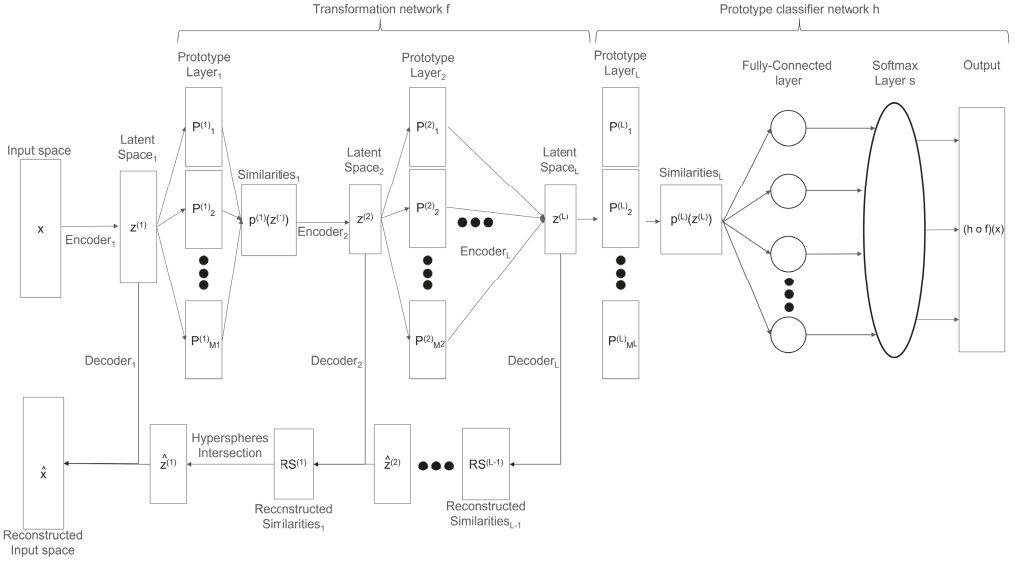


Figure 1. Deep Prototype-Based Network architecture.

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \subset \mathcal{X} \times \{1, \ldots, K\}$, $\mathcal{X} \subset \mathbb{R}^p$ denote a finite training set sampled from an underlying distribution $\mathbb{P}_{XY}$. All spaces are Euclidean and, therefore, equipped with the canonical inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\| \cdot \|_2$. The uppercase calligraphic symbols note sets or spaces, and the superscripts in parentheses index the depth $\ell \in \{1, \ldots, L\}$. The set of all network parameters is denoted as $\Theta = \{\theta_{f^{(1)}}, \{\theta_{e^{(\ell)}}, \theta_{d^{(\ell)}}\}_{\ell=2}^{L}, \{P_j^{(\ell)}\}, W\}$.

3.1. The Forward Encoding Flow

Figure 1 shows that the forward flow begins in the input space, followed by the first encoder, which maps the input data to a latent space. Then, the transformation network f comprises L-1 stacked triples consisting of a prototype layer, the calculation of a similarity vector, and an encoder for the following layer. Finally, the prototype classifier network h takes as input the last prototype layer (L), calculates the similarity vector, and passes the results through a linear mapping with a subsequent softmax layer that produces the estimated classification distribution of an input.

The input space domain $\mathcal{X}$ is a subset of a p-dimensional real vector space. A sample is denoted by $\mathrm{x} \in \mathcal{X}$. No further structure is assumed, except for the differentiability of all subsequent mappings defined on $\mathcal{X}$.

The first layer of the DPBN is the first encoder, which projects input data onto a lower-dimensional manifold:

$$f^{(1)} : \mathcal{X} \longrightarrow \mathcal{Z}_1, \qquad \mathcal{Z}_1 = \mathbb{R}^{q_1}, \quad q_1 \ll p,$$

$$f^{(1)}(\mathrm{x}) = z^{(1)}.$$

The mapping $f^{(1)}$ is usually implemented using a convolutional neural network that ends with a fully connected layer of width q_1. The parameters associated with this mapping are collectively denoted by $\theta_{f^{(1)}}$ and form part of the general parameter set.

The next layer is the first prototype layer $\mathcal{P}^{(1)}$,

$$\mathcal{P}^{(1)} = \{P_j^{(1)}\}_{j=1}^{M_1} \subset \mathcal{Z}_1,$$

and comprises a finite set of prototypes. The value $M_1 \in \mathbb{N}$ is a fixed hyperparameter a priori, while prototypes are learned during the training process. Given a latent representation $z^{(1)}$, its similarity vector is calculated using the squared Euclidean distance to all prototypes. It is collected in the similarity vector

$$p^{(1)}(z^{(1)}) = \left(d_1^{(1)}(z^{(1)}), \ldots, d_{M_1}^{(1)}(z^{(1)})\right)^\top \in \mathbb{R}^{M_1}, d_j^{(1)}(z^{(1)}) = \|z^{(1)} - P_j^{(1)}\|_2^2,$$

which defines M_1 differentiable functions $d_j^{(1)} : \mathcal{Z}_1 \to \mathbb{R}^+$, thereby enabling the application of gradient-based optimization methods without requiring any modifications.

The next layer is the second encoder. The resulting similarity vector $p^{(1)}(z^{(1)})$ computed in the previous layer, which does not contain any positional ordering, is compressed by a learned distance encoder to obtain the second layer representation.

$$e^{(2)} : \mathbb{R}^{M_1} \longrightarrow \mathcal{Z}_2, \qquad \mathcal{Z}_2 = \mathbb{R}^{q_2},$$

$$z^{(2)} = e^{(2)}\big(p^{(1)}(z^{(1)})\big).$$

The function $e^{(2)}$ is implemented as a multilayer perceptron, with its parameters collectively denoted by $\theta_{e^{(2)}}$. The representation $z^{(2)}$ can be expressed as $z^{(2)} = e^{(2)} \circ p^{(1)} \circ f^{(1)}(\mathrm{x})$ using the composition of functions.

Similarly to the first layer, for every subsequent layer $\ell \in \{2, \ldots, L\}$, there is an encoder that generates the latent space $\mathcal{Z}_\ell$, followed by a prototype set $\mathcal{P}^{(\ell)} = \{P_j^{(\ell)}\}_{j=1}^{M_\ell} \subset \mathcal{Z}_\ell$, then a similarity vector $p^{(\ell)}(z^{(\ell)})$, and finally an encoder for the next layer $\ell + 1$:

$$p^{(\ell)}(z^{(\ell)}) = \left(d_1^{(\ell)}(z^{(\ell)}), \ldots, d_{M_\ell}^{(\ell)}(z^{(\ell)})\right)^\top \in \mathbb{R}^{M_\ell},$$

$$e^{(\ell+1)} : \mathbb{R}^{M_\ell} \longrightarrow \mathcal{Z}_{\ell+1}, \qquad z^{(\ell+1)} = e^{(\ell+1)}\big(p^{(\ell)}(z^{(\ell)})\big).$$

We denote as $\Phi^{(\ell)}(\mathrm{x}) = e^{(\ell)} \circ p^{(\ell-1)} \circ e^{(\ell-1)} \circ \cdots \circ p^{(1)} \circ f^{(1)}(\mathrm{x}) = z^{(\ell)}$ the composite mapping that transforms x into $z^{(\ell)}$.

To provide the final classification decision after passing through all layers of the DPBN, the final similarity vector $p^{(L)}\big(z^{(L)}\big) \in \mathbb{R}^{M_L}$ is fed into a linear map $W \in \mathbb{R}^{K \times M_L}$ to obtain

$$\eta(x) = W\, p^{(L)}\big(z^{(L)}\big) \in \mathbb{R}^K.$$

The subsequent softmax activation function produces the estimated class distribution $\widehat{y}(\mathrm{x})$, where $\widehat{c}(\mathrm{x}) = \arg\max \widehat{y}(\mathrm{x})$ is the predicted class.

A summary of the forward flow from the input vector x to the predicted class distribution is as follows:

$$\mathrm{x} \xrightarrow{f^{(1)}} z^{(1)} \xrightarrow{\mathcal{P}^{(1)}} p^{(1)}(z^{(1)}) \xrightarrow{e^{(2)}} z^{(2)} \xrightarrow{\mathcal{P}^{(2)}} \cdots \xrightarrow{e^{(L)}} z^{(L)} \xrightarrow{\mathcal{P}^{(L)}}$$

$$p^{(L)}(z^{(L)}) \xrightarrow{W+Softmax} \widehat{y}.$$

3.2. The Backward Decoding Flow

What distinguishes Prototype-Based Networks, and in particular the proposed DPBN, from regular neural networks is the presence of prototypes that, through their reconstruction back into the input space, offer visual explanations for their predictions. However, this visualization is not straightforward in the case of the DPBN. The procedure to visualize deeper prototypes introduces a novel approach called the hypersphere intersection (HS-Int) algorithm, which transforms a reconstructed similarity vector at deep ℓ, $\mathrm{RS}^{(\ell)}$ in Figure 1 or $\widetilde{p}^{(\ell)}$, into a reconstructed code $\widetilde{z}^{(\ell)}$.

Figure 1 also illustrates the backward decoding flow, which reconstructs the deepest code $z^{(L)}$ back into the input space. This reconstruction process performs L-1 decoding and hypersphere intersection pairs until the first reconstructed code $\widetilde{z}^{(1)}$ is obtained. Finally, $\widetilde{z}^{(1)}$ is decoded back into $\widetilde{\mathrm{x}}$.

The hypersphere intersection algorithm is based on previous results from algebraic and geodesy/GPS positioning [15–17], as well as nonlinear least-squares optimization (Levenberg–Marquardt) [18,19] and the Moore–Penrose generalized inverse matrix theory [20,21].

For any depth $\ell \geq 2$, the decoder produces the radius vector

$$\widehat{p}^{(\ell-1)} = (\widehat{\rho}_1^{(\ell-1)}, \ldots, \widehat{\rho}_{M_{\ell-1}}^{(\ell-1)})^\top, \quad \widehat{\rho}_j^{(\ell-1)} \geq 0,$$

which, together with the prototype set $\mathcal{P}^{(\ell-1)} = \{P_j^{(\ell-1)}\}_{j=1}^{M_{\ell-1}} \subset \mathcal{Z}_{\ell-1}$, defines $M_{\ell-1}$ hyperspheres in the ambient space $\mathcal{Z}_{\ell-1}$ as follows:

$$\mathcal{S}_j = \{z \in \mathcal{Z}_{\ell-1} : \|z - P_j\|_2^2 = \widehat{\rho}_j^2\},$$

where $P_j = P_j^{(\ell-1)}$ and $\widehat{\rho}_j^2 = \left(\widehat{\rho}_j^{(\ell-1)}\right)^2$ to simplify the notation. Thus, the objective becomes to recover a point $z^\star \in \bigcap_{j=1}^{M_{\ell-1}} \mathcal{S}_j$, or, if the intersection is empty, the solution to the minimization problem:

$$z^\star = min_{z \in \mathcal{Z}_{\ell-1}}\{F(z)\}, \qquad F(z) = \sum_{j=1}^{M_{\ell-1}} \left(\|z - P_j\|_2^2 - \widehat{\rho}_j^2\right). \tag{1}$$

We draw on previous studies in this field to design the algorithm in three steps. The first is to perform linear reduction by applying Bancroft's algebraic difference trick [15] to obtain the linear system:

$$Az = b, \qquad A \in \mathbb{R}^{(M_{\ell-1}-1)\times q_{\ell-1}},\ b \in \mathbb{R}^{M_{\ell-1}-1}.$$

Its least-squares solution is $z_p = A^+ b$, where $= A^+$ is the Moore–Penrose pseudoinverse. Every other solution differs from z_p by an element of the null space of A:

$$z \;=\; z_p + N\alpha, \qquad N \in \mathbb{R}^{q_{\ell-1}\times d},\ \alpha \in \mathbb{R}^d,\ d = q_{\ell-1} - \operatorname{rank}(A),$$

where $\operatorname{rank}(A)$ represents the rank of matrix A, N stacks an orthonormal basis of $\ker A$, with $\ker A$ being the kernel of matrix A, so d equals the number of remaining degrees of freedom and α are their coordinates. If $d = 0$, the linear part has isolated at most one candidate; otherwise, the quadratic constraint of the next step must fix α (or show that there is no exact intersection).

The second step inserts the above-mentioned parameterization $z = z_p + N\alpha$ into the first sphere equation $\|z - P_1\|_2^2 = \widehat{\rho}_1^2$, which yields

$$q(\alpha) \;=\; \|\, z_p + N\alpha - P_1\|_2^2 - \widehat{\rho}_1^2 = 0.$$

If $d = 0$, no free parameters remain, and q reduces to a constant; the unique linear candidate z_p is an exact intersection; otherwise, no actual solution exists.

If $d = 1$, then $q(\alpha)$ is an ordinary quadratic in one variable. Its roots have closed forms tabulated in [17] producing up to two exact points, which we collect in $\mathcal{Z}_{\text{exact}}$.

Lastly, if $d \geq 2$, then it means that the same equation defines at most a $(d-1)$-dimensional quadric in the α space. Therefore, we launch a Levenberg–Marquardt search [19] from $\alpha = 0$ to find a feasible α (or certify that none exists), following the least-squares recipe in [17].

Finally, any α that satisfies $q(\alpha) = 0$ is mapped through $z = z_p + N\alpha$ and added to $\mathcal{Z}_{\text{exact}}$ in the third step. However, if no exact intersection is found, ($\mathcal{Z}_{\text{exact}} = \varnothing$), we minimize the nonlinear least-squares cost

$$F(z) \;=\; \tfrac{1}{2} \sum_{j=1}^{M_{\ell-1}} \big(\, \|z - P_j\|_2^2 - \widehat{\rho}_j^2\big)^2.$$

The optimization process uses the Levenberg–Marquardt-damped Gauss–Newton scheme [18,19], which smoothly interpolates between gradient descent and the Gauss–Newton step via an adaptive damping factor. After convergence, we obtain $z^\star$. The local uniqueness is verified with the Jacobian rank criterion of Abel-Chaffee [16]. The algorithm finally returns $\widetilde{z}^{(\ell-1)} = \mathrm{z}^\star$ together with the uniqueness flag.

Once the HS-Int algorithm is described, we can define the decoding cascade, which starts from $z^{(L)}$, applies a decoding step

$$\widehat{p}^{(\ell-1)} = d^{(\ell)}\big(z^{(\ell)}\big),$$

and then applies the HS-Int algorithm

$$\widetilde{z}^{(\ell-1)} = \text{HS-Int}\big(\mathcal{P}^{(\ell-1)}, \widehat{p}^{(\ell-1)}\big)$$

to $\ell = L, L-1, \ldots, 2$. Note that the decoding step transforms a code from a latent space into a reconstructed similarity vector. Then, the HS-Int algorithm takes the reconstructed similarity vector and the set of prototypes from the corresponding prototype layer to generate the reconstructed code for the previous layer. Finally, $\widetilde{\mathrm{x}} = d^{(1)}\big(z^{(1)}\big) \in \mathcal{X}$ is the global reconstruction against which the reconstruction term in (2) is measured. This cascade enforces explicit geometric consistency between successive latent spaces, endowing every latent point with a deterministic pre-image in the input domain.

The reconstruction pathway can be summarized as

$$z^{(L)} \xrightarrow{d^{(L)}} \widehat{p}^{(L-1)} \xrightarrow{\text{HSInt}} \widetilde{z}^{(L-1)} \xrightarrow{d^{(L-1)}} \cdots\ \widetilde{z}^{(1)} \xrightarrow{d^{(1)}} \widetilde{\mathrm{x}}.$$

The proposed DPBN provides expressive capacity and interpretability. Every depth $\ell \geq 2$ applies the composition $e^{(\ell)} \circ p^{(\ell-1)}$, which is Lipschitz continuous when $e^{(\ell)}$ employs bounded activation functions, endowing the network with a hierarchical piecewise-smooth warping of the latent metric. Yet, interpretability is preserved: each prototype $P_j^{(\ell)}$ is deterministically mapped to the input domain by the reconstruction chain, which yields a visual or otherwise perceptible exemplar representing that prototype at its innate semantic granularity.

3.3. Guided Prototype Learning

Together with the pipelines for classification and decoding, it is crucial to define the joint learning objective of the network architecture.

Let $\mathcal{L}_{\mathrm{CE}}(\widehat{y}, y) = -y \log \widehat{y}$ be the cross-entropy between the prediction $\widehat{y}$ and the ground truth y. With $\{z_i^{(\ell)}\}_{i=1}^{B}$ referring to the latent codes of a mini-batch of size B, we define the layer-wise prototype coverages as

$$R_1^{(\ell)} = \frac{1}{M_\ell} \sum_{j=1}^{M_\ell} \min_{i=1,\ldots,B} \left\| P_j^{(\ell)} - z_i^{(\ell)} \right\|_2^2, \qquad R_2^{(\ell)} = \frac{1}{B} \sum_{i=1}^{B} \min_{j=1,\ldots,M_\ell} \left\| z_i^{(\ell)} - P_j^{(\ell)} \right\|_2^2.$$

Similarly, with a mini-batch of size B, we establish the loss functions of the autoencoders for the image and the distances as

$$\mathcal{L}_{\mathrm{img}}^{\mathrm{AE}} = \frac{1}{B} \sum_{i=1}^{B} \left\| \widetilde{x}_i - x_i \right\|_2^2 \qquad \mathcal{L}_{\mathrm{dist}}^{\mathrm{AE}} = \frac{1}{B} \sum_{i=1}^{B} \sum_{\ell=2}^{L} \left\| \widetilde{d}_i^{(\ell)} - d_i^{(\ell)} \right\|_2^2.$$

As such, for positive scalars acting as regularization terms $\lambda_{\mathrm{R_I}}$, $\lambda_{\mathrm{R_D}}$, $\lambda_{\mathrm{R}_1}^{(\ell)}$, and $\lambda_{\mathrm{R}_2}^{(\ell)}$, the total loss reads

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}_{\mathrm{CE}}\big(\widehat{y}(x_i), y_i\big) + \lambda_{\mathrm{R_I}} \mathcal{L}_{\mathrm{img}}^{\mathrm{AE}} + \lambda_{\mathrm{R_D}} \mathcal{L}_{\mathrm{dist}}^{\mathrm{AE}} + \sum_{\ell=1}^{L} \Big(\lambda_{\mathrm{R}_1}^{(\ell)} R_1^{(\ell)} + \lambda_{\mathrm{R}_2}^{(\ell)} R_2^{(\ell)} \Big). \tag{2}$$

All model parameters Θ are trained by stochastic gradient descent on (2), with gradients obtained via automatic differentiation through the analytic Jacobian of HS-Int whenever the exact intersection is unique; otherwise, the sub-gradient of the best-fit residual is used.

4. Experiments and Results

In this section, we conduct a systematic evaluation of the proposed Deep Prototype-Based Network, complemented by a series of targeted architectural variants. This analysis aims to demonstrate the model's capabilities while also addressing a key challenge: the tension between visually interpretable prototype reasoning and the internal mechanisms of class assignment within the network.

The first is a baseline experiment to assess performance on the MNIST dataset [4] and to examine the prototype representations obtained from the proposed DPBN architecture (see Figure 1). The second experiment is a model distillation study in which we investigate whether the prototypes learned by the deepest layer of a four-layer DPBN can be effectively transferred to a significantly shallower model, without compromising either predictive performance or interpretability. The third experiment addresses a binary classification task designed to demonstrate that the network can produce accurate predictions and visualizations consistent with the class "0", while its internal explanatory mechanisms align more closely with class "1".

For each experimental configuration, we report both classification accuracy and a novel validation metric introduced in this study—the Normalized Negative Entropy (NNE) score. Quantitative findings are further supported by qualitative analyses, comprising visualizations of reconstructed prototypes and learned weight matrices. Collectively, the results elucidate the impact of each architectural choice on the reliability, interpretability, and overall performance of DPBNs.

All the experiments are conducted using the MNIST [4] benchmark dataset. The original training set, comprising 60,000 images, is randonmly partitioned into 90% for training and 10% for validation. The official test set of 10,000 images is held out for evaluation. All images are resized to 28×28 pixels and normalized to the $[0, 1]$ range using min-max scaling.

4.1. NNE Score

Prototype-Based Networks typically conclude with a linear classifier, where the weight matrix $W \in \mathbb{R}^{K \times M_L}$ maps prototype–sample distances to class logits. To assess the interpretability of this final classification layer, we introduce the NNE score. This metric interprets the prototype-to-class weight matrix W as an information-theoretic channel. After negating W, a row-wise softmax is applied to obtain a class probability distribution for each prototype. The Shannon entropy of each distribution is then computed, inverted, normalized by $\log K$, and averaged across all prototypes. The resulting NNE score lies within the interval $[0, 1]$, attaining a maximum value of 1 only when each prototype exhibits a one-hot (i.e., perfectly class-specific) voting pattern.

In contrast to geometric alignment measures—such as the B-Cos transform [22], which enhances the fidelity of saliency maps by maximizing cosine similarity between inputs and classifier weights, or the support-vector-based alignment strategy employed by WASUP [23]—the NNE metric exclusively captures the distributional sharpness of the classification head, providing a complementary perspective on model interpretability.

Because large negative distances should yield a high vote, we first negate the weight matrix and apply a row-wise softmax:

$$P = \sigma(-W^\top) \in [0,1]^{M_L \times K}, \qquad \textstyle\sum_{k=1}^{K} P_{ik} = 1 \;\; \forall\, i.$$

Each row p_i can now be read as a categorical distribution over classes for prototype i. As such, for a row p_i, we can compute its Shannon entropy

$$H(p_i) \;=\; -\sum_{k=1}^{K} p_{ik} \, \log p_{ik}, \tag{3}$$

and normalize it by the maximum possible entropy $\log K$ to obtain a confidence score

$$\tilde{H}(p_i) \;=\; 1 - \frac{H(p_i)}{\log K}, \quad \tilde{H}(p_i) \in [0,1]. \tag{4}$$

$\tilde{H}(p_i) = 1$ when p_i is a one-hot vector (prototype decisively represents one class) and $\tilde{H}(p_i) = 0$ when p_i is uniform (prototype conveys no class information).

Finally, we average over all P prototypes:

$$\mathrm{NNE}(W) \;=\; \frac{1}{P} \sum_{i=1}^{P} \tilde{H}(p_i). \tag{5}$$

$\mathrm{NNE} \in [0, 1]$ is therefore *higher* for more interpretable weight matrices. We validated the metric with two extreme cases as shown at Table 1.

Table 1. Extreme matrices and their corresponding normalized negative entropy (NNE) values.

Matrix	**NNE**
Identity $\mathbf{I}_K$ (*perfectly sharp*)	1.000
Uniform $\left(\frac{1}{K}\mathbf{1}\right)$ (*fully diffuse*)	0.000

It complements structural sparsity metrics such as global/local size and Hoyer–Square norms in ProtoS-ViT [24] and the "transparent-head complexity" of Shallow-ProtoPNet [25] by quantifying decisiveness even when the head is dense. Where neuron-level Class-Selectivity Indices normalize entropy over activations [26], NNE is the first to apply an entropy normalization directly to prototype voting weights, making it agnostic to input distribution and layer depth. Recent analyses of classifier weight spectra, enabled by bilinear multilayer perceptrons (MLPs) [27], underscore the value of interpreting the classification head as an information-theoretic channel. In this context, the NNE score provides a concise, task-size-invariant metric that aligns with and supports this perspective.

The proposed metric will be employed throughout this work to evaluate the quality of the classification weight matrix across all experiments, serving as a means to assess and compare the reliability and class specificity of the learned prototypes.

4.2. Experiment 1: Baseline Architecture

The objective of this experiment is twofold: to evaluate the performance of the proposed architecture on the MNIST dataset and to analyze the structure of the resulting prototype representations. As an initial step, we consider the canonical DPBN, which closely follows the configuration introduced by [2], with minor modifications to facilitate subsequent ablation studies. The input pipeline preserves the original 28×28 luminance pixel values of the MNIST digits, thereby avoiding information loss during preprocessing.

Within the network, layer 1 comprises compact convolutional encoder–decoder module that projects each input image into a 40-dimensional latent vector and reconstructs it back into pixel space. The encoder employs a fixed stride pattern of $(2, 2, 2, 2)$ and a sequence of convolutional filters $(32, 32, 32, 10)$, progressively reducing the spatial resolution from 28×28 to 2×2 over four downsampling stages. This architectural design ensures that each spatial location in the final feature map corresponds to a roughly square receptive field in the input image, a property leveraged in later experiments to interpret prototype activations in a spatially meaningful manner.

The flattened feature map is subsequently projected into a latent representation of dimension $q_1 = 15$ via a fully connected layer. The choice of $q_1 < 40$ is deliberate, as it constrains the image autoencoder to learn a non-trivial, low-dimensional manifold rather than approximating the identity mapping. This serves as an implicit regularization mechanism for the initial prototype search space. Within this space, $M_1 = 20$ prototypes are allocated, resulting in a prototype-to-latent ratio of $M_1/q_1 = 1.33$. Empirical observations from preliminary experiments indicated that this ratio is sufficient to support a near one-to-one correspondence between digit classes and prototypes, while avoiding the introduction of redundant representations.

The subsequent layers (layers 2 through 4) share an identical structural design, differing from the initial layer primarily in that their encoders are implemented as MLPs operating on similarity vectors rather than raw image data. Specifically, each distance vector produced by layer l is projected into a latent space of dimension $q_{l+1} = 10$, then decoded back to its original dimensionality M_l, and subsequently passed to the next prototype layer of equal size. In each layer, the prototypes $p_j^{(l)}$ are initialized from a uniform distribution $\mathcal{U}(0, 1)$ and jointly optimized alongside the corresponding autoencoder weights. Importantly, we enforce the condition $M_l \geq q_l$ at every depth to ensure that the resulting

distance matrices are overcomplete—a necessary property for the hypersphere intersection solver used to reconstruct prototype representations across layers.

The coefficients we set to $\lambda_{R_I} = \lambda_{R_D} = \lambda_{R_1} = \lambda_{R_2} = 0.5$ based on a coarse log-grid search aimed at balancing classification accuracy with reconstruction fidelity. Model training was performed using the Adam optimizer with a fixed learning rate of 1×10^{-4}. An early stopping criterion with a patience of 100 epochs was employed, terminating training when validation accuracy failed to improve over 100 consecutive epochs. On the MNIST dataset, the model converged to a test accuracy of 98.05% after approximately 850 epochs, thereby establishing a strong performance upper bound for the ablation studies that follow.

Interpretability, however, diminishes with increasing network depth. As illustrated in Figure 2, reconstructed prototypes from all four layers are visually compared. Prototypes in the first layer exhibit clearly defined glyphs, each distinctly associated with a specific digit class. This observation confirms that the autoencoder, in conjunction with the R_1/R_2 regularization terms, effectively aligns prototypes with the underlying data manifold. By layer 2, the prototypes lose fine-grained details and become more abstract, although they still retain rough class-specific contours. In layers 3 and 4, interpretability degrades substantially: the reconstructed shapes converge into nearly indistinguishable forms, differentiated only by subtle intensity variations and with weakly class-informative interpretable patterns. Despite this loss in human interpretability, the softmax-normalized weight matrix yields a NNE score of 0.26, indicating that the classifier maintains a coherent and class-specific internal representation, even as the prototypes themselves become less visually discernible. This observation is particularly significant, as it indicates that the final-layer prototypes retain sufficient information to support accurate classification while simultaneously integrating multiple visual patterns. This behavior is a consequence of the decreasing dimensionality across successive layers, which compels the network to fuse lower-level features. Accordingly, the process can be interpreted as a hierarchical compositional mechanism that preserves discriminative attributes across layers.

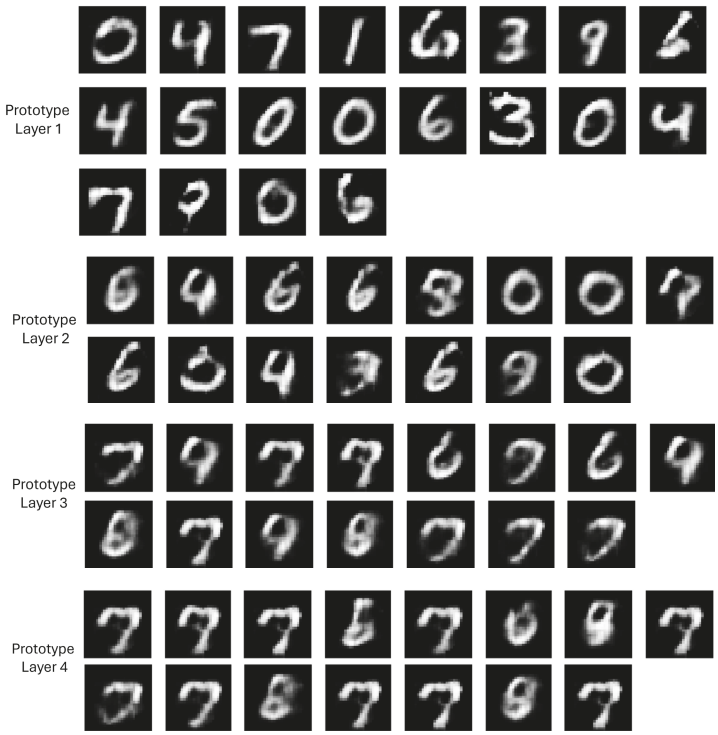


Figure 2. Reconstructions of prototypes reveal that the uppermost panel (layer 1) exhibits digit-specific structures, while deeper layers (from top to bottom) tend to converge toward patterns that are less interpretable by humans due to their combination.

4.3. Experiment 2: Model Distillation

In this experiment, we investigate whether the prototypes learned by the deepest layer of a four-layer DPBN could be effectively transferred to a significantly shallower model without compromising either predictive accuracy or interpretability. Specifically, the prototype layer extracted from the fourth layer of the reference DBPN was copied into a single-layer DPBN, where it remained fixed during training. Only the autoencoder and the classification head were subsequently trained. Qualitative comparisons between the original layer 4 prototypes and those reconstructed by the distilled model, along with the associated weight matrices W_1 and W_2, are shown in Figures 3 and 4.

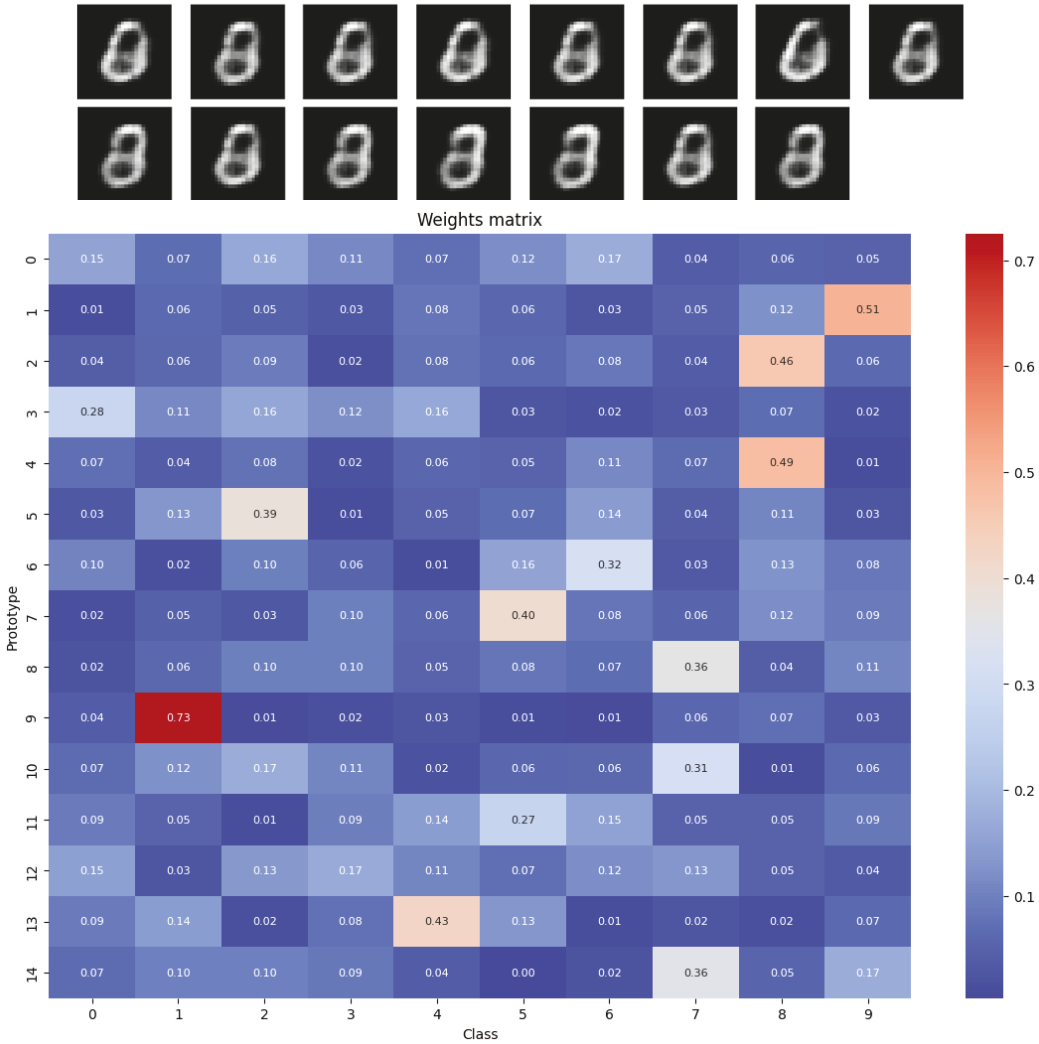


Figure 3. Prototypes derived from the final layer in Experiment 2, presented alongside the corresponding weight matrix.

Despite the substantial reduction in network depth, the distilled model achieved an accuracy statistically indistinguishable from that of the baseline. Moreover, its weight matrix maintained coherence, with an NNE of 0.18. To assess whether this performance could be attributed to the information encoded in the transferred prototypes, we repeated the experiment using randomly initialized prototype vectors, which were likewise frozen during training. Contrary to our initial hypothesis, the network again attained comparable accuracy and produced sharply defined reconstructed prototypes. This suggests that the raw numerical values of the prototypes, in isolation, contribute little semantic content. Rather, interpretability appears to arise from the dynamics of the autoencoder and the spatial configuration of the prototypes within the latent space. Notably, the geometry of

this space retains semantic structure: prototypes corresponding to visually similar digits, such as 8 and 9, are positioned closely, whereas those representing semantically distant digits, such as 2 and 9, are more widely separated. It is also worth noting that incorrect associations appear in the weight matrix (Figure 4), where prototype 12 (row 11) is assigned to class '0', whereas it can be visually interpreted as an '8' or '9'. We postulate that this latent space organization underlies the richer structure observed in the weight matrix when training begins with learned or trainable prototypes as opposed to random ones. These findings motivated the follow-up experiment described in the subsequent section, in which the network was trained end-to-end using randomly initialized, frozen prototypes. This design aimed to isolate the contribution of latent space geometry from the influence of prototype initialization.

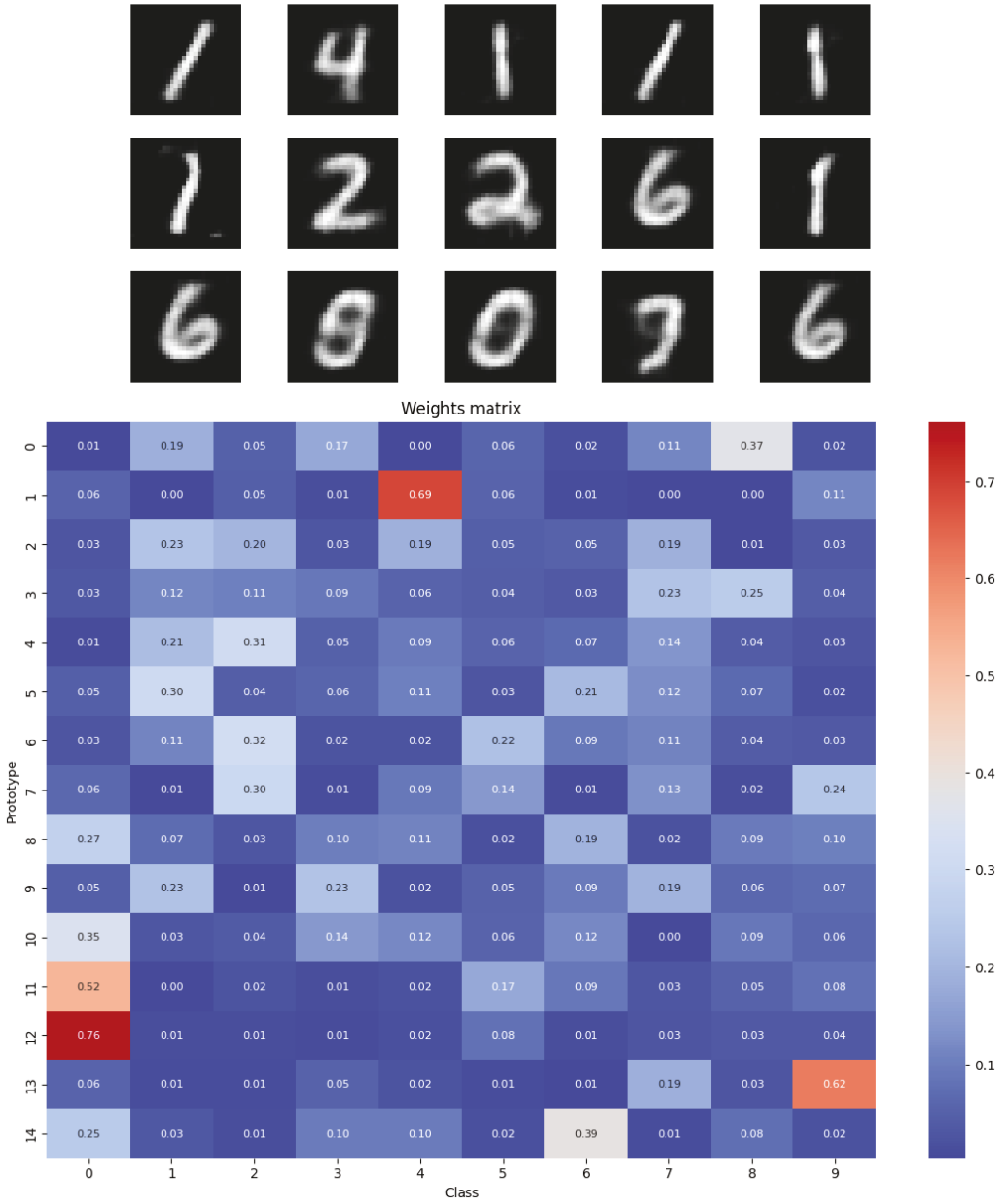


Figure 4. Prototypes obtained from those transferred from a 4-layer DPBN in Experiment 2, shown together with the corresponding weight matrix. The prototypes are indexed by row and column from 0 to 14.

4.4. Experiment 3: Can a 1 Look like a 0?

In this experiment we investigate whether the conclusions drawn in the previous multiclass setting generalize to the simpler binary task of distinguishing the digit "0" from all other handwritten digits. To construct the training corpus, all images corresponding to class "0" from the MNIST training split were retained, while samples from the remaining classes were randomly downsampled to ensure that the alternative class (denoted other) contained an equal number of instances. The resulting dataset was thus balanced, with 80% allocated for training, 10% for validation, and the remaining 10% for testing. All target labels were transformed into binary form using the indicator function $y \mapsto 1[y \neq 0]$.

We retained the deep prototype architecture of the previous experiments, but specified a four-layer prototype hierarchy $(15, 20) \to (10, 15) \to (10, 15) \to (10, 4)$, where each pair denotes the latent dimensionality, and the number of prototypes. The network was trained for 370 epochs with Adam (learning rate = 10^{-4}) and early stopping after 100 epochs without validation improvement. The highest validation accuracy achieved was 99.49%, with a final test accuracy of 99.48%. Despite this strong predictive performance, the interpretability metrics were less encouraging. Specifically, the softmax-normalized weight matrix of the final classification layer exhibited a NNE of 0.14, indicating a relatively flat association between prototypes and class labels. Furthermore, two of the four prototypes reconstructed from the deepest layer resembled noisy, amorphous "clouds" (see Figure 5) rather than well-defined digit patterns, and the weight matrix (see Figure 6) deviated substantially from a one-hot configuration. Regarding class purity or faithfulness, Figure 6 shows a faithfulness of $1/4$, where all prototypes are assigned to class 1, but prototype 2 is correctly assigned to 0. This example is representative of the results achieved with other instances, which have been tested under human supervision.

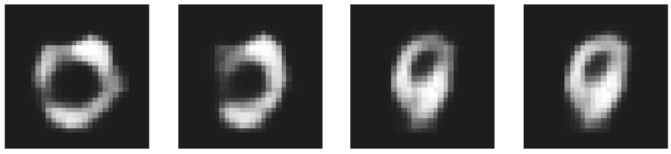


Figure 5. Reconstructed prototypes from the last layer showing a low degree of interpretability.

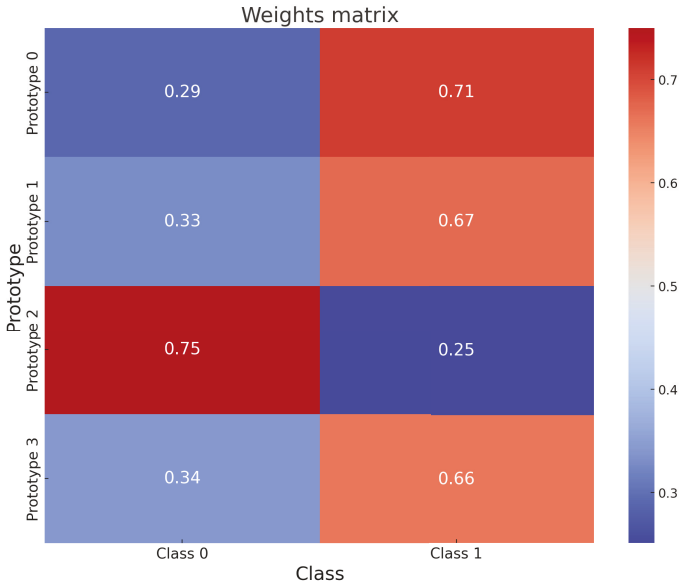


Figure 6. Softmax-normalized weight matrix of the Deep Prototype Network. Rows correspond to prototypes, columns to the two classes. Prototype #4 (bottom row) is visually a digit "0" yet assigns the bulk of its weight to the other class.

To examine whether the latent geometry captured by the deep model could be more explicit, we distilled the last-layer prototypes into a shallow DPBN comprising a single layer, following the same procedure as in Experiment 2. The prototypes were held fixed during training, while only the autoencoder and classification weights were updated. Training converged after 210 epochs, achieving an accuracy of 99.83%, marginally surpassing that of the original deep model. These results support the hypothesis that a well-structured set of prototypes can facilitate more efficient representation learning, even when the prototypes themselves lack of clear visual interpretability.

A detailed examination of the distilled model revealed an noteworthy behavior. Prototype #4, whose reconstruction clearly depicts a digit "0" (see Figure 7), is associated by the weight matrix with the 'others' class with a probability of 66 %. An analysis of the twelve test images with latent encodings closest to this prototype (Figure 8) confirms that all correspond to well-formed instances of the digit "0", each correctly classified as class "0". Consequently, the prototype provides clear visual evidence of the digit "0" while functionally contributing to the decision for the opposing class. Because the mapping between prototypes and classes is learned independently of the visual characteristics of the prototypes, users lacking access to the weight matrix may be misled into believing that the model's predictions for the "others" class are based on archetypal non-zero digits, when in fact, the decisive signal is anchored in a visually canonical zero.

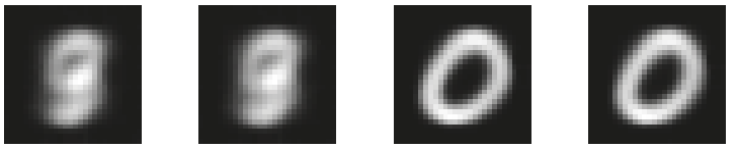


Figure 7. Reconstructed deepest-layer prototypes after training. The last prototype (right-most) is a clear "0" despite its functional association with the other class.

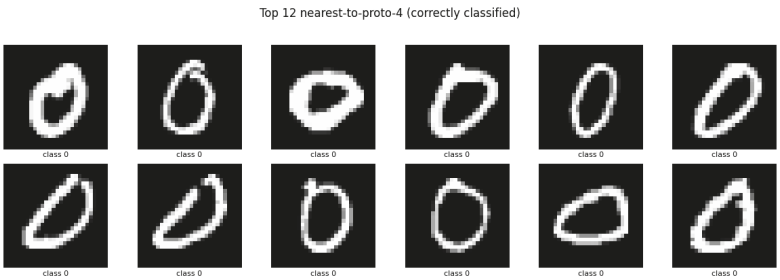


Figure 8. Twelve test images nearest to Prototype #4 in latent space, all correctly labeled as class "0".

The binary classification setting thus amplifies a concern previously suggested by the multiclass experiments: prototype visualizations alone are insufficient to establish model trustworthiness. A prototype may function as a decoy, simultaneously presenting a compelling, human-interpretable exemplar while driving predictions for an unexpected class. Given that the distance vectors input to the linear classification head are high-dimensional, manual verification of class–prototype associations is impractical, and summary metrics such as NNE reflect only global sharpness rather than individual inconsistencies.

5. Conclusions

In this study, we introduced a novel prototype-based neural network architecture incorporating multiple prototype layers. We evaluated its performance and interpretability using the MNIST dataset, demonstrating both its representational capabilities and highlighting critical limitations concerning interpretability. Results demonstrate the network's capacity to integrate and represent patterns across successive prototype layers. However, we observed a progressive decline in the visual interpretability of prototypes at deeper

layers, complicating human understanding of the learned representations. Importantly, the decline in visual clarity at deeper prototype layers did not compromise the model's predictive performance, which remained consistently high—indicating that the final prototype layer retained strong discriminative capacity despite reduced interpretability. A distilled shallow model, constructed by reusing the final prototype layer from the original DPBN as a fixed component, achieved a slight improvement in classification accuracy (99.83%). However, it exposed a critical flaw—prototypes visually representing "0" contributed internally to decisions for the "1" class. Hence, visualizations of prototypes alone are insufficient for model transparency and reliable interpretability, and it is required to align visual and functional behaviors. Accordingly, we argue that prototype-based explanations within the "this-looks-like-that" paradigm necessitate additional safeguards—either through training constraints that enforce alignment between visual and functional semantics, adjusting design parameters (such as number of prototypes), or via post hoc auditing tools capable of identifying contradictory associations. A comprehensive investigation into such reliability guarantees is reserved for future work.

Author Contributions: Conceptualization, E.G.-C.; methodology, E.G.-C. and D.M.; software, R.C.I.; validation, E.G.-C., D.M. and R.C.I.; formal analysis, E.G.-C. and D.M.; investigation, E.G.-C., R.C.I., and D.M.; data curation, R.C.I.; writing—original draft preparation, E.G.-C. and R.C.I.; writing—review and editing, E.G.-C. and D.M.; visualization, R.C.I.; supervision, E.G.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Stat. Surv.* **2022**, *16*, 1–85. [CrossRef]
2. Li, O.; Liu, H.; Chen, C.; Rudin, C. Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
3. Ras, G.; Xie, N.; Van Gerven, M.; Doran, D. Explainable deep learning: A field guide for the uninitiated. *J. Artif. Intell. Res.* **2022**, *73*, 329–396. [CrossRef]
4. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
5. Chen, C.; Li, O.; Tao, D.; Barnett, A.; Rudin, C.; Su, J. This Looks Like That: Deep Learning for Interpretable Image Recognition. In Proceedings of the NeurIPS, Vancouver, BC, Canada, 8–14 December 2019.
6. Stefenon, S.F.; Singh, G.; Yow, K.C.; Cimatti, A. Semi-ProtoPNet Deep Neural Network for the Classification of Defective Power Grid Distribution Structures. *Sensors* **2022**, *22*, 4859. [CrossRef] [PubMed]
7. Singh, G.; Yow, K.C. Object or Background: An Interpretable Deep Learning Model for COVID-19 Detection from CT-Scan Images. *Diagnostics* **2021**, *11*, 1732. [CrossRef]
8. Singh, G. One and one make eleven: An interpretable neural network for image recognition. *Knowl.-Based Syst.* **2023**, *279*, 110926. [CrossRef]
9. Nauta, M.; van Bree, R.; Seifert, C. Neural Prototype Trees for Interpretable Fine-Grained Image Recognition. In Proceedings of the CVPR, Nashville, TN, USA, 19–25 June 2021.
10. Gulshad, S.; Long, T.; van Noord, N. Hierarchical Explanations for Video Action Recognition. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 18–22 June 2023; pp. 3703–3708. [CrossRef]
11. Koh, P.W.; Nguyen, T.; Tang, Y.S.; Mussmann, S.; Pierson, E.; Kim, B.; Liang, P. Concept Bottleneck Models. In Proceedings of the ICML, Virtual Event, 13–18 July 2020.
12. Papernot, N.; McDaniel, P. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. In Proceedings of the ICML, Stockholm, Sweden, 10–15 July 2018.

13. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic Routing Between Capsules. In Proceedings of the NeurIPS, Long Beach, CA, USA, 4–9 December 2017.
14. Zhang, N.; Donahue, J.; Girshick, R.; Darrell, T. Part-based R-CNNs for Fine-grained Category Detection. In Proceedings of the ECCV, Zurich, Switzerland, 6–12 September 2014.
15. Bancroft, S. An Algebraic Solution of the GPS Equations. *IEEE Trans. Aerosp. Electron. Syst.* **1985**, *AES-21*, 56–59. [CrossRef]
16. Abel, J.; Chaffee, J. Existence and Uniqueness Analysis for the GPS Equations. *IEEE Trans. Aerosp. Electron. Syst.* **1991**, *27*, 952–956. [CrossRef]
17. Norrdine, A. An Algebraic Solution to the Multilateration Problem. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia, 13–15 November 2012; pp. 1–6.
18. Levenberg, K. A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Q. Appl. Math.* **1944**, *2*, 164–168. [CrossRef]
19. Marquardt, D.W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441. [CrossRef]
20. Moore, E.H. On the Reciprocal of the General Algebraic Matrix. *Bull. Am. Math. Soc.* **1920**, *26*, 394–395.
21. Penrose, R. A Generalized Inverse for Matrices. *Proc. Camb. Philos. Soc.* **1955**, *51*, 406–413. [CrossRef]
22. Bohle, M.; Singh, N.; Fritz, M.; Schiele, B. B-Cos Alignment for Inherently Interpretable CNNs and Vision Transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 4504–4518. [CrossRef] [PubMed]
23. Wolf, T.N.; Kavak, E.; Bongratz, F.; Wachinger, C. SIC: Similarity-Based Interpretable Image Classification with Neural Networks. *arXiv* **2025**, arXiv:2501.17328.
24. Turbé, H.; Bjelogrlic, M.; Mengaldo, G.; Lovis, C. ProtoS-ViT: Visual foundation models for sparse self-explainable classifications. *arXiv* **2024**, arXiv:2406.10025.
25. Singh, G.; Frizzo Stefenon, S.; Yow, K.C. The shallowest transparent and interpretable deep neural network for image recognition. *Sci. Rep.* **2025**, *15*, 13940. [CrossRef]
26. Leavitt, M.L.; Morcos, A. Selectivity considered harmful: evaluating the causal impact of class selectivity in DNNs. *arXiv* **2020**, arXiv:2003.01262. [CrossRef]
27. Pearce, M.T.; Dooms, T.; Rigg, A.; Oramas, J.M.; Sharkey, L. Bilinear MLPs enable weight-based mechanistic interpretability. *arXiv* **2025**, arXiv:2410.08417.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

From Games to Understanding: Semantrix as a Testbed for Advancing Semantics in Human–Computer Interaction with Transformers

Javier Sevilla-Salcedo *, José Carlos Castillo Montoya, Álvaro Castro-González and Miguel A. Salichs

RoboticsLab, Universidad Carlos III de Madrid, 28911 Madrid, Spain; jocastil@ing.uc3m.es (J.C.C.M.), acgonzal@ing.uc3m.es (Á.C.-G.), salichs@ing.uc3m.es (M.A.S.)
* Correspondence: jasevill@ing.uc3m.es

Abstract: Despite rapid progress in natural language processing, current interactive AI systems continue to struggle with interpreting ambiguous, idiomatic, and contextually rich human language, a barrier to natural human–computer interaction. Many deployed applications, such as language games or educational tools, showcase surface-level adaptation but do not systematically probe or advance the deeper semantic understanding of user intent in open-ended, creative settings. In this paper, we present Semantrix, a web-based semantic word-guessing platform, not merely as a game but as a living testbed for evaluating and extending the semantic capabilities of state-of-the-art Transformer models in human-facing contexts. Semantrix challenges models to both assess the nuanced meaning of user guesses and generate dynamic, context-sensitive hints in real time, exposing the system to the diversity, ambiguity, and unpredictability of genuine human interaction. To empirically investigate how advanced semantic representations and adaptive language feedback affect user experience, we conducted a preregistered 2 × 2 factorial study (N = 42), independently manipulating embedding depth (Transformers vs. Word2Vec) and feedback adaptivity (dynamic hints vs. minimal feedback). Our findings revealed that only the combination of Transformer-based semantic modelling and adaptive hint generation sustained user engagement, motivation, and enjoyment; conditions lacking either component led to pronounced attrition, highlighting the limitations of shallow or static approaches. Beyond benchmarking game performance, we argue that the methodologies applied in platforms like Semantrix are helpful for improving machine understanding of natural language, paving the way for more robust, intuitive, and human-aligned AI approaches.

Keywords: semantic understanding; human–computer interaction; transformer models; natural language processing; user engagement; interactive systems

1. Introduction

Enabling machines to comprehend human language's full richness, ambiguity, and flexibility remains a central challenge in artificial intelligence and human–computer interaction (HCI). While recent advances in natural language processing (NLP) have spurred impressive progress, the question of how to systematically evaluate and deepen machine semantic understanding, beyond surface-level syntax to context, nuance, and communicative intent, remains unresolved. Addressing this gap calls for innovative testbeds that expose systems to unpredictable, creative, and open-ended human inputs, requiring real-time interpretation and adaptation.

In this work, we introduce Semantrix, an interactive semantic word-guessing game designed not merely for entertainment but as a platform to probe and accelerate the development of machine semantic understanding. The game situates state-of-the-art Transformer models in a dynamic, user-driven environment where every user's guess and system response become a data point for the real-time study of ambiguity, context, and adaptive language processing. In Semantrix, machine learning models must not only score and evaluate user guesses for semantic proximity but also generate personalised, context-sensitive hints, requiring both deep comprehension and generative skill.

Word games and puzzle-based interfaces have a long tradition in cognitive science as tools for investigating the capacity of humans, and, by extension, artificial agents, to reason, associate, and communicate meaning under uncertainty. We move beyond static benchmarking and toward ecological, interactive assessment by embedding modern language models within such a framework. Whereas previous NLP evaluation tasks have typically relied on fixed datasets or limited response formats, the gameplay context elicits a wide variety of free-form, user-driven language, a crucible for examining machine robustness to error, ambiguity, and creativity. Beyond text-only contexts, recent advances in adaptive representation learning show how meta-learning can improve robustness under domain shift. For example, CAMeL introduces cross-modality adaptive meta-learning for text-based person retrieval, emphasising task diversity and domain-agnostic pretraining to generalise from biased synthetic image–text data to real benchmarks; this underscores the value of adaptation strategies that we pursue here in an interactive, user-facing context [1].

The rise of Transformer architectures such as BERT [2] and GPT-4 [3] has dramatically expanded the capabilities of NLP systems to encode, compare, and generate natural language. Yet, it remains to be seen how these models perform in environments that demand fine-grained semantic judgement, such as assessing the "distance" between two words or ideas and generating instructive, meaningful feedback. Building on successes in grounded language learning [4] and the emerging literature on educational and gamified AI interfaces [5,6], Semantrix merges these two threads: robust semantic modelling and adaptive, generative dialogue in an ecologically valid, user-facing scenario.

Despite these advances, contemporary systems still grapple with interpretive failures: idiomatic, ambiguous, or contextually entangled inputs frequently derail dialogue or hinder effective feedback. In word-guessing games, such challenges are amplified, as users often generate intentionally indirect, metaphorical, or creatively oblique guesses. An effective system must do more than parse tokens; it must infer meaning, respond adaptively, and know when and how to scaffold user learning with timely, personalised hints.

Semantrix thus serves as a living laboratory that supports several interconnected research objectives. First, it enables the evaluation of contemporary Transformer-based models concerning their capacity to model and compare semantic content at the word and short-phrase level, all within realistic, unconstrained contexts. Second, the platform allows one to investigate how generative models can adaptively support user understanding and engagement throughout the interaction when used for dynamic hint production. Finally, Semantrix provides a unique opportunity to study the interplay between algorithmic advances and human factors such as motivation, engagement, and strategy that naturally emerge in digital language games. The platform leverages Sentence Transformers for robust multilingual semantic scoring, alongside large language models such as GPT-4 for generating context and performance-sensitive hints. Extensive logging and experimental controls permit analysis and ablation across model types, hints strategies, and user behaviours. From an HCI standpoint, unified experience models offer a complementary lens on what interactive AI should optimise. A recent unified model for Haptic Experience synthesises pragmatic, hedonic, and eudaimonic qualities to guide the design of technology-mediated

touch. However, our modality is textual/visual; this framework highlights the importance of designing for experiential richness. It suggests directions for multimodal hints and accessibility features in future iterations of Semantrix [7].

This paper claims that interactive, game-based frameworks such as Semantrix do not merely entertain; they create new frontiers for studying and refining machine semantic understanding in the wild. The insights from such environments extend beyond gaming, illuminating pathways for advancing more intuitive, adaptive, and human-aware NLP systems across domains.

The remainder of this paper is structured as follows. Section 2 reviews the state of the art in semantic embedding and generative language models. Section 3 details the NLP models, natural language generation technologies, and web-deployment frameworks employed in Semantrix. Section 4 presents the whole design of the semantic guessing game, including game mechanics, feedback processes, and the adaptive hint generation system. Section 5 describes the integration, architecture, and deployment of the Gradio-based web application. Section 6 outlines the methodology, experimental design, and participant recruitment. Section 7 reports the behavioural outcomes and statistical analyses. Section 8 discusses the results regarding interpretation, limitations, and future directions. Finally, Section 9 summarises this work's main contributions and implications.

2. Background: Semantic Embedding and Generative Models

Understanding and operationalising semantic similarity remains a core challenge in NLP and HCI. Semantic word games, such as the one addressed in this work, demand models that can assess proximity between user guesses and secret targets based on minimal input (typically a single word or short phrase), and in multilingual or code-switching contexts. This section surveys classic and state-of-the-art alternatives, offering a comparative perspective and motivating the final selections for our application.

Early approaches to semantic similarity leveraged static distributed embeddings, most notably Word2Vec [8] and GloVe [9]. These models represent each word as a fixed vector, efficiently capturing distributional relationships from large corpora and supporting many languages. However, assigning each word a single embedding regardless of context fails to resolve polysemy or ambiguity, a significant limitation for applications like word games where user inputs can be brief and heavily context-dependent. Furthermore, while multilingual variants exist, their cross-lingual capabilities are typically weaker and less robust, particularly for short or ambiguous queries.

The advent of Transformer-based architectures, such as BERT [2], RoBERTa [10], MPNet [11], and their multilingual derivatives (e.g., XLM-RoBERTa [12]), has enabled a new level of contextualisation in language representations. Embeddings generated by these models are sensitive to the surrounding text, better capturing the nuance required for semantic similarity tasks. Adaptations like Sentence-BERT (SBERT) [13] further adapt these models for short-text or sentence-level comparison, providing more reliable polysemy resolution and significantly improving robustness to ambiguity. Moreover, many of these Transformers have been fine-tuned for word- or sentence-level similarity across multiple languages, which is crucial for mixed-language user contexts. Notably, they lead recent benchmarks such as the Massive Text Embedding Benchmark (MTEB) [14] for tasks relevant to our setting.

In our survey of candidate models for semantic similarity, we considered Word2Vec, for its efficiency and interpretability, although acknowledging its lack of contextual sensitivity, as well as a range of Transformer-based models, including SBERT and paraphrase-multilingual-mpnet-base-v2, which offer advanced contextualisation and strong multilingual support, as demonstrated by recent MTEB results. Additional Transformer variants,

such as RoBERTa, XLM-R, and MPNet, with or without further fine-tuning, were also evaluated for potential applicability. Word2Vec and Transformers were compared empirically to assess their impact on user experience and behaviour in the experimental phase.

Beyond modelling semantic similarity, providing real-time, contextually relevant feedback and hints is central to interactive word games. Rule-based or manually scripted responses often prove insufficient: the deterministic nature of pre-written hints restricts the system's capacity to adapt to unanticipated user actions, making it challenging to offer practical assistance in dynamic scenarios. Moreover, scripted strategies possess limited creativity and variation, which experienced users may exhaust or learn to predict, ultimately reducing engagement and replayability.

The emergence of large autoregressive language models (LLMs), developed by organisations such as OpenAI (e.g., GPT-3, GPT-4 [3]), Google (Gemini 2.5 [15]), Mistral [16], and others, has transformed the capabilities of natural language generation. In numerous languages and formats, these models can produce flexible, high-quality text outputs for various content types, including hints, definitions, analogies, and curiosity facts. Critically, they generate personalised, context-sensitive responses that can dynamically adapt to user actions and learning trajectories, providing supportive feedback that evolves with user experience. This high naturalness has substantially increased user engagement and the perceived value of in-game assistance.

Our pilot testing showed that GPT-4 delivered significant qualitative and practical improvements for dynamic, multilingual, and challenging hint-generation tasks, outperforming earlier models and numerous open-source alternatives. Nevertheless, our selection process also weighed strong contenders from other providers, such as Google Gemini and Mistral's advanced LLMs, which offer competitive performance and diverse deployment options.

The final system was thus constructed around two main components: for semantic similarity calculation, we selected paraphrase-multilingual-mpnet-base-v2 [13] due to its robustness with short inputs and strong multilingual capabilities; for natural language generation, we adopted GPT-4, balancing high output quality with API accessibility despite higher resource demands. Importantly, our modular platform design preserves flexibility for future experimentation, including potential integration of alternative LLMs, such as Gemini, Mistral, or DeepSeek, as their capabilities evolve.

3. Materials

In exploring the design and implementation of a semantic word-guessing game, we focus on leveraging modern natural language processing techniques within a web-based interface. The primary objective of this section is to present the technical framework, detailing our integration of Transformer-based models for semantic similarity evaluation and dynamic hint generation, all accessed through an interactive Gradio application.

In the Semantrix application, semantic similarity between user guesses and target words is computed using the publicly available paraphrase-multilingual-mpnet-base-v2 Sentence Transformer model, which provides robust, contextualised embeddings suitable for real-time gameplay. No additional fine-tuning was performed; the model is used as released for efficient extraction of semantic relationships from user input, supporting multilingual interaction and accurate feedback throughout the game. The application currently supports English and Spanish. For each language, we provide UI text, rules and word lists. To add another language, one can supply its word list, and, if needed, adjust the UI text and rules in configuration files. No source-code changes are required.

Central to our approach is the use of Gradio (https://www.gradio.app/ (accessed on 24 May 2025), version 5.21.0), a Python-based framework that enables the development of

interactive machine learning interfaces. The complete application is deployed on Hugging Face Spaces (https://huggingface.co/spaces (accessed on 24 May 2025)), which provides the hosting infrastructure and integrates seamlessly with Gradio to maximise accessibility and maintainability for the research team and the user community. This combination offers several practical advantages for developing and disseminating NLP-powered applications. First, Gradio supports rapid prototyping and iterative model integration, making developing, refining, and evaluating interactive systems with state-of-the-art NLP components straightforward. Deploying the application on Hugging Face Spaces further streamlines the process, as updates and new model versions can be pushed live with minimal overhead, eliminating the need for dedicated server maintenance or complex configuration.

The resulting platform is accessible from any modern web browser without installation or user registration, lowering participation barriers and facilitating broad experimental reach. Its design is responsive and inherently multiplatform, ensuring a consistent user experience across devices such as smartphones, tablets, and desktop computers, and automatically adapting to different screen sizes and input methods.

Regarding reproducibility and open science, the deployment on Hugging Face Spaces allows our semantic game to remain version-controlled and openly accessible, supporting independent review, reuse, and further extension by the research community. Additionally, Gradio's integration with the Hugging Face model repository and backend infrastructure simplifies the technical implementation and ensures that the interface components are closely aligned with the underlying model resources.

Overall, this architecture provides a robust, modular foundation for ongoing experimentation and future extension, facilitating the rapid translation of NLP research advances into accessible, interactive applications.

The core mechanism for gameplay, measuring semantic proximity between user guesses and the secret word, requires a model that can produce highly expressive and context-robust word embeddings. Unlike sentence-level similarity, the word-guessing context is challenging: user input is brief, often a single word.

Our approach employs Sentence Transformers, a bi-encoder architecture wherein a Transformer model encodes both target and guess in parallel. Figure 1 illustrates the bi-encoder architecture used for evaluating semantic similarity between words in our system. In this setup, each input word, the target word (Word A) and the user's guess (Word B), is independently processed by an identical encoder model. The models encode each input word, generating contextualised token representations. These representations are subsequently transformed into fixed-size sentence embeddings via a pooling operation (such as taking the [CLS] token or averaging all token embeddings), resulting in two dense vectors, denoted u and v. Finally, the semantic similarity between these two vectors is computed using cosine similarity. The resulting score quantifies the semantic proximity between the guess and the target word, enabling the system to provide meaningful feedback based on the user's input. This architecture allows for efficient and scalable evaluation, as both words can be encoded in parallel, and is well suited for real-time applications like interactive word games.

For simplicity, throughout this paper, we refer to paraphrase-multilingual-mpnet-base-v2 as Sentence Transformers.

Semantic scoring is performed with an encoder-only model, but hint generation requires dynamic, natural-sounding output. This needs a decoder-only, autoregressive language model. Encoder models capture "meaning"; decoder models generate "language" in context. For tailored hints, clues, definitions, and "curiosity facts," we require a model to interpret prompts flexibly and produce suitable out-of-context responses.

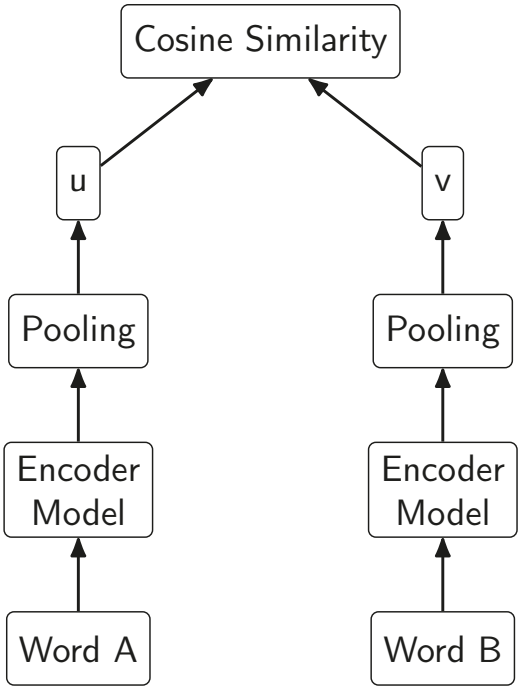


Figure 1. Bi-encoder architecture for semantic word similarity measurement.

All generation tasks are handled by OpenAI's GPT-4 and accessed via an API. Key attributes:

- Foundation: Pretrained on enormous web, book, and dialogue corpora; further aligned with human feedback [3].
- Capabilities: Multilingual, robust in general and domain-specific query answering, creativity (emoji, poetry), and varied response formality.
- Integration: Prompt templates specify output language, type, and difficulty. The model generates hints in response to gameplay events.
- Limits: Online inference only, subject to latency and API reliability.

All system components, including embedding models, hint generation, session management, and user feedback, are integrated within a unified Gradio interface and deployed on Hugging Face Spaces. This setup provides a convenient, cross-platform web application accessible via any browser, enabling broad user participation for experimental studies. The combination of Gradio and Hugging Face simplifies deployment and model updates, supports reproducible experiments through automated logging and data collection, and allows easy maintenance and rapid prototyping as research evolves. Moreover, the public deployment facilitates transparency and open science, as the system code and configuration are available for review and reuse by other researchers. This infrastructure ensures a robust foundation for our experiments and supports future platform extensions, including multimodal or robotic interface integration.

4. Game Design

In this section, we aim to present the overall design of the game experience, detailing its structure and individual modules, which are elaborated upon later. The game's main objective is for the user to guess a secret word. Initially, the user may attempt different words randomly, but their guesses should become more refined as the game progresses; this way, a secondary objective is to make this process of knowledge refinement mentally stimulating for the user.

4.1. Initialisation

The interaction flow of the game has been structured to provide a seamless and engaging user experience, as illustrated in Figure 2. Upon initialising the game, as described in the first row of the Figure (green section), users are greeted with an introduction screen. They are then offered the option to review the game's rules; this is particularly important for first-time users or those who need a refresher. If they choose to review the rules, the system provides a detailed description before proceeding to the game setup.

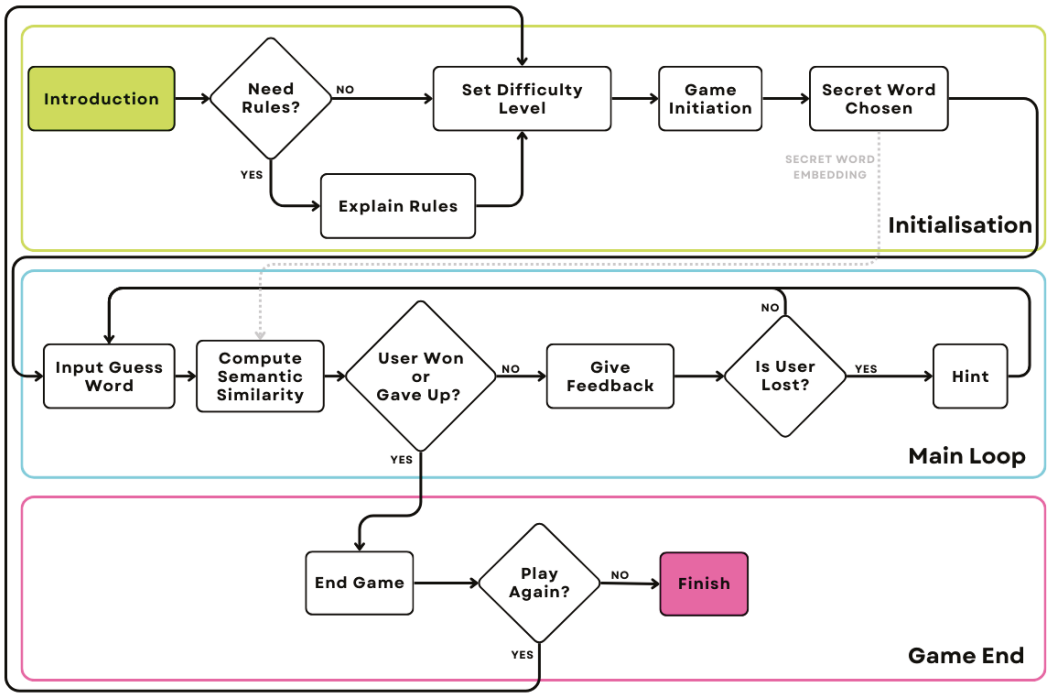


Figure 2. Flow diagram of the game performance. Initialisation introduces the game, optionally explains the rules, sets the difficulty level, starts the game, and selects the secret word (its embedding is stored; the dashed arrow indicates its use in the similarity calculation). In the main loop, the user enters a guess word, we calculate its semantic similarity to the secret word and branch as follows: if the user has won or given up, the loop ends; otherwise, we provide feedback and, if the user is stuck, give them a hint before the next guess. Finally, the user can start another round or end the game.

Once past the introductory phase, users can select the difficulty level, a feature crucial for determining the secret word's complexity and the nature of hints provided. Our system offers four difficulty levels: easy, normal, hard, and expert. When users choose the easy or normal level, the secret word is drawn from a basic vocabulary list, making the game accessible to a broad audience. In contrast, selecting hard or expert levels requires users to guess words from a more complex, domain-specific vocabulary, thus increasing the challenge. All word lists have been curated from language learning resources and classified according to their semantic complexity.

The selected difficulty level also governs the hint system: on easy mode, users only receive advanced hints, while expert mode offers no hints. Players receive the standard initial and advanced hints sequence for normal and hard modes. A detailed description of hint types is provided below.

This design ensures the game adapts to varying user knowledge and preferences by modulating both the lexical complexity of the target word and the level of support provided via hints. After users configure their preferred settings, the system randomly selects a secret word from the appropriate list and generates its embedding vector to optimise further game processes.

4.2. Main Loop: Semantic Understanding

With all preparations complete, the main game loop begins, as illustrated in the second row (blue section) of Figure 2. The process starts when the user inputs a guess (Input Guess Word). We then Compute Semantic Similarity between that guess and the secret word using cosine similarity; the dashed link from Secret Word Chosen in Figure 2 indicates the reuse of the precomputed embedding of the secret word at this step. Each guessed word is converted into an embedding vector, allowing the game to compute the semantic similarity between the guessed and pre-selected secret words. This similarity score generates feedback, guiding the user towards the correct answer. This feedback mechanism is central to assisting users throughout their gameplay experience.

To generate the similarity score, each word undergoes processing by the Sentence Transformer model, which produces an embedding vector. The specific model employed generates vectors with a dimensionality of 768. Once embedding vectors for the guessed and the secret words have been obtained, the similarity score is calculated using the cosine similarity measure. Cosine similarity was chosen due to its widespread use and effectiveness in comparing high-dimensional semantic embeddings: it quantifies the cosine of the angle between two vectors, thereby capturing semantic closeness based on their orientation in the embedding space, independent of their magnitude. We derive the cosine similarity value from the dot product equation, as illustrated in Equation (1). We can isolate the vectors from this equation, enabling us to compute the cosine similarity. The dot product equation is mathematically expressed as:

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i \tag{1}$$

where $\vec{a}$ and $\vec{b}$ are the vectors representing the guessed word and the secret word, respectively, and n is the dimensionality of these vectors. To transition from the dot product to the cosine similarity, we utilise Equation (2)

$$\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \tag{2}$$

where $\cos(\theta)$ denotes the cosine similarity, $\vec{a} \cdot \vec{b}$ is the dot product, and $\|\vec{a}\|$ and $\|\vec{b}\|$ are the magnitudes of vectors $\vec{a}$ and $\vec{b}$, respectively. The magnitudes are computed as follows:

$$\|\vec{a}\| = \sqrt{\sum_{i=1}^{n} a_i^2}$$

$$\|\vec{b}\| = \sqrt{\sum_{i=1}^{n} b_i^2}$$

By substituting these magnitudes back into the cosine similarity formula, we obtain Equation (3).

$$\cos(\theta) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}} \tag{3}$$

We derive the cosine value through this transformation. Then, once multiplied by ten, we obtain a semantic score over ten, quantifying the semantic similarity between the guessed and secret words. This quantified measure of similarity forms the foundation of our feedback. Similarly, all the scores obtained during each round are recorded to analyse the user's performance within the hint system. This data collection also serves the secondary purpose of informing the user about the words they have attempted during the

game that are closest to the secret word. This allows users to leverage this information in the game's feedback mechanism. Tracking these scores can give users detailed insights into their guessing patterns and improve the gameplay experience. The stored data help refine the hint system and ensure that users can learn from previous attempts, enhancing their strategy with each subsequent guess.

Initially, if the user's guess is far from the secret word, the system provides encouraging but non-specific feedback, indicating whether they are "hot" or "cold" in semantic proximity. The feedback becomes increasingly targeted as the guesses draw closer to the secret word. While the feedback does not directly give hints about the word, it operates with the dynamic hint system that offers progressively specific clues when appropriate, narrowing down the available options and aiding the user in their quest to identify the secret word. In the web application, feedback is conveyed through the graphical user interface and on-screen messages, rather than by physical devices or speech. All system outputs, including feedback and hints, are presented visually within the web interface. Because input is unrestricted, users often can try metaphorical or slang words. In this context, every guess is embedded and scored identically against the secret word; the interface immediately shows the numerical score and qualitative "hot/cold" feedback and updates the best-tries table. When a sequence of such guesses shows a sustained downward trend according to the detector, which the extended use of ambiguous words could cause, the system gives a hint that narrows the semantic search space without revealing the answer.

4.3. Dynamic Hint System

Identifying the optimal moment to provide hints to the user presents a nuanced challenge, as there is a risk of the user perceiving the assistance as either offensive or condescending. Bearing this in mind, our objective is to offer subtle assistance when we detect that the user is experiencing difficulty or is stagnating without being intrusive. This approach ensures that the user remains engaged and enjoys the experience.

As previously mentioned, data collection enables us to analyse the user's performance throughout their interaction with the game. To effectively capture trends and fluctuations in user performance, we adopted the Exponential Moving Average (EMA), a widely used technique in performance analysis across fields such as sports [17] and finance [18]. The first EMA applied in our system uses a window of five samples (n = 5), corresponding to a smoothing factor of (α = 0.33), as shown in Equation (4) below. This EMA provides a smoothed representation of the user's score progression by assigning exponentially greater weight to more recent observations, making it more responsive to current changes than a simple arithmetic mean. This improved sensitivity to recent data points allows the system to detect trends in user performance more rapidly, helping us provide timely and context-appropriate hints.

$$\mathrm{EMA}_t = \alpha \cdot \mathrm{P}_t + (1 - \alpha) \cdot \mathrm{EMA}_{t-1} \tag{4}$$

where EMA_t is the Exponential Moving Average at time t; α is the smoothing factor, defined as $\alpha = \frac{2}{n+1}$ with n being the number of samples considered; P_t is the observed value (e.g., performance score) at time t; and EMA_{t-1} is the EMA from the previous period.

The application of the EMA allows us to smooth out irregularities and short-term fluctuations in performance, while retaining sufficient sensitivity to recent user actions. This balance is essential for adaptive educational interventions, such as calibrating the timing and specificity of game hints. Figure 3 illustrates this process. In the upper half of the figure, the black dots represent individual user scores for each guess, which can fluctuate substantially from one attempt to the next. The blue line shows these scores' EMA, providing a smoother trajectory that reveals the underlying trend in user performance,

rather than being dominated by noise or outliers. For example, even though there may be sharp drops or spikes in individual scores, the EMA curve responds more gradually, making it easier to distinguish between random variation and sustained changes in performance.

Beyond smoothing, it is crucial to determine the direction and rate of change in performance to identify optimal moments for intervention. To this end, we evaluate the slope of the Exponential Moving Average, which reflects whether user performance is improving, declining, or stable. The slope is estimated by computing the derivative of the EMA, using the gradient formula detailed in Equation (5).

$$\mathrm{M_{EMA}} = \frac{\partial \mathrm{EMA}}{\partial t} = \frac{\Delta \mathrm{EMA}}{\Delta t} \tag{5}$$

Given the slope of user performance in each game iteration, we can observe a trend in the performance metrics. However, these metrics are notably volatile and significantly influenced by game iteration changes. To address this, we further smooth the slope signal by applying a second EMA to the derivative, using a window of three samples (n = 3), corresponding to a smoothing factor of (α = 0.5). This approach provides a more nuanced smoothing, balancing capturing immediate changes and mitigating aggressive spikes in our metrics. The outcomes of our approach are illustrated in the lower part of Figure 3, which displays the derivative (slope) of the EMA (green squares), alongside an EMA-smoothed estimate of the derivative itself (red diamonds). When the slope is close to zero, user performance is stable. Positive slope values signal improving performance, while negative slope values suggest declining performance. For instance, in guesses 5 to 9, the negative slope indicates a decline in performance, which could be an optimal point to trigger an intervention or provide a targeted hint. Conversely, after guess 13, the trend reverses, reflecting a rapid improvement captured by the raw and smoothed slope curves.

We must establish a threshold that acts as the trigger to leverage these data to determine the optimal moment to provide hints to the user. This threshold was defined heuristically and derived from testing with several users. We must establish a threshold based on trends extracted from the moving average statistics to determine the optimal moment to provide hints to the user. Specifically, the system monitors the Exponential Moving Average (EMA, $\alpha = 0.5$) of the first discrete derivative of the score EMA ($\alpha = 0.33$). A hint is triggered when this derived value falls below zero, when user performance exhibits a sustained downward trend, indicating the user is likely struggling or stagnating. When this condition is met, the interface shows the hint panel. To avoid premature intervention, the system requires at least three consecutive guesses to have occurred since the last hint before considering a new one. Furthermore, due to the initial EMA computation with a window of five, the system will only begin evaluating these thresholds and offering its first hint after a minimum of five user guesses.

This combined threshold mechanism was defined heuristically based on pilot experiments with several users. As a result, we can robustly identify when a user is lost or not making progress solely based on performance metrics. The next step involves devising dynamic and creative strategies to assist the user, ensuring that help is valuable and engaging, while maintaining the integrity of the game. Consequently, our initial problem is effectively addressed; we can now identify when a user is lost or stagnate solely based on their performance metrics. The next step involves devising strategies to assist the user dynamically and creatively, ensuring that the help is valuable and entertaining while not compromising the game experience. This requires a carefully balanced approach to give hints or guidance that enhances the user's engagement and enjoyment without spoiling the core challenges of the game.

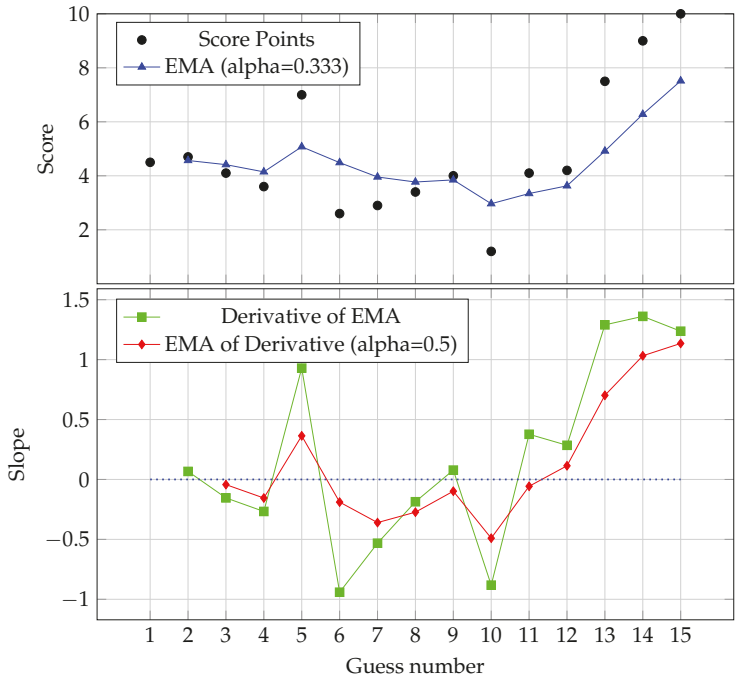


Figure 3. Visualization of user guess scores (dots), the Exponential Moving Average (EMA, blue), the first discrete derivative of the EMA (green), and the EMA of the derivative (red) over 15 guesses with $\alpha_{\text{EMA}} = 0.333$ and $\alpha_{\text{EMA-deriv}} = 0.5$. This sequence allows the system to track performance trends and identify moments for adaptive hint intervention.

Given this framework, we meticulously designed a set of dynamic hints categorised into two distinct tiers. The first tier consists of subtle cues that provide slight indications regarding the target word, assisting users in navigating the semantic theme without directly revealing the solution. Conversely, the second tier comprises more explicit and conspicuous hints. While these clues do not directly disclose the answer, they significantly narrow down the possibilities, guiding the user towards an imminent discovery.

In addition to hint categories, as mentioned above, our system adapts to the initial difficulty level the user selects. For moderate difficulty, the system is more likely to provide second-tier hints when users encounter impediments. At higher difficulty levels, users predominantly receive first-tier hints unless prolonged stagnation is detected, in which case, more advanced hints may be provided. In the "expert" mode, users face all challenges entirely unaided, receiving no hints during gameplay.

The dynamic hint system incorporates various hint types based on natural language processing and generative models. Table 1 summarises these hint categories, describes their purpose, and indicates the model(s) used and the type of output involved. For added clarity, concrete examples of each kind of hint are provided immediately after the table.

Examples of hints for the secret word "Apple" :

- Word Ranking: e.g., "fruit: 92%, car: 15%, school: 54%, blue: 35%." The user sees a ranked list expressing the semantic closeness of various words to the secret word.
- Film Representation: e.g., 👱‍♀️🍎🪞👑👵💤🏰🧚‍♀️👸🤴🧙‍♀️🧛‍♀️🎁🧜‍♀️🧞‍♂️🪶🌲 (a sequence of emojis referencing the film "Snow White").
- Word Definition: e.g., "A fruit with sweet red or green skin and crisp flesh."
- Word Representation: e.g., 🍎 for "apple" (an emoji or pictogram representation of the word).
- Poem: e.g., "Red or green, crisp and sweet, Harvest joy you love to eat."

Table 1. Classification of dynamic hint types used in the system, including models involved and type of output.

Title	Description	Models Used	Type of Output
Initial Hints			
Word Ranking	Semantic rank of four random words	Sentence Transformers and GPT-4	Image
Film Representation	Representation of a film related to the word using only emojis	GPT-4	Image
Advanced Hints			
Word Definition	Definition of the word	GPT-4	Text
Word Representation	Representation of the word using only emojis	GPT-4	Image
Poem	Poem about the secret word	GPT-4	Text

As detailed previously in Section 3, given the unpredictability of the secret word selection, our system dynamically generates hints irrespective of the specific word while ensuring consistent structural patterns across different games. We utilise autoregressive models, particularly those optimised for text generation tasks such as GPT models, to achieve this. We designed dedicated processes for each predefined hint structure that combine state-of-the-art semantic and generative models. For example, to generate the "Word Ranking" hint, we select three random words and one semantically close to the secret word, as determined by the Sentence Transformers embedding model. These four words are then presented to the user in a ranked list, indicating their semantic proximity to the secret word. For other hint types such as "Word Definition", "Film Representation", "Poem", or "Word Representation", we employ carefully designed, task-specific prompts for each hint type, submitting them directly to GPT-4, which generates the corresponding output (e.g., definitions, relevant emoji sequences, or creative texts). This modular approach enables dynamic, context-appropriate, and engaging hint generation for any secret word, while ensuring consistency in structure and user experience throughout the game.

Although we adopted an EMA-based trigger, other policies are reasonable. Rule-based triggers (e.g., triggering after a fixed number of failed attempts, or when scores remain below a threshold) are simple and transparent. Still, they may ignore the magnitude and direction of change and require frequent readjustments, sometimes triggering after harmless oscillations. Time-based triggers are easy to implement but they combine reflection time, writing speed, and device or network latency, so they may not reliably indicate that a player has been lost. On the other hand, user-initiated hints maximise autonomy and transparency but can introduce biases that complicate comparisons between conditions; in the pilot tests where we enabled the option to request hints, several participants avoided asking for help even when they got stuck, which increased the variability in frustration. In contrast, our EMA-based policy is lightweight and reacts to trends rather than noise or peaks. The two-stage EMA described above reduces fluctuation while maintaining sensitivity to avoid getting stuck. It still requires parameter choice and a brief warm-up period.

4.4. Game End

The game concludes when the participant either accurately identifies the secret word or opts to withdraw. Upon conclusion, users are presented with a comprehensive performance summary detailing the number of attempts taken, the overall accuracy of their guesses, and a curiosity about the secret word, which is dynamically generated via a language model. This feature enhances the application's interactive and educational attributes, providing users with a sense of accomplishment and motivating continued play.

A key aspect for sustaining user engagement is increasing the game's replayability. To this end, users are prompted to start a new game with a new secret word at the end

of each session. This mechanism introduces variety and unpredictability, encouraging repeated playthroughs and maintaining user interest. The game's design aims to balance challenge and approachability, offering an engaging experience that appeals to many users.

5. Integration in the Semantrix Web Application

The Semantrix word association game was developed as a highly modular and configurable web application, enabling research and public deployment scenarios. The web-based architecture supports dynamic experimental conditions, session management, multilingual interaction, and robust data handling, all operated through an intuitive user interface built with the Gradio framework. Figure 4 presents an overview of the Semantrix web application architecture, highlighting the modular front-end, backend, session management, configuration, and data flows described in detail in the following sections. Arrows indicate interaction and data flow; coloured backgrounds are used solely to distinguish component groups, without further visual meaning.

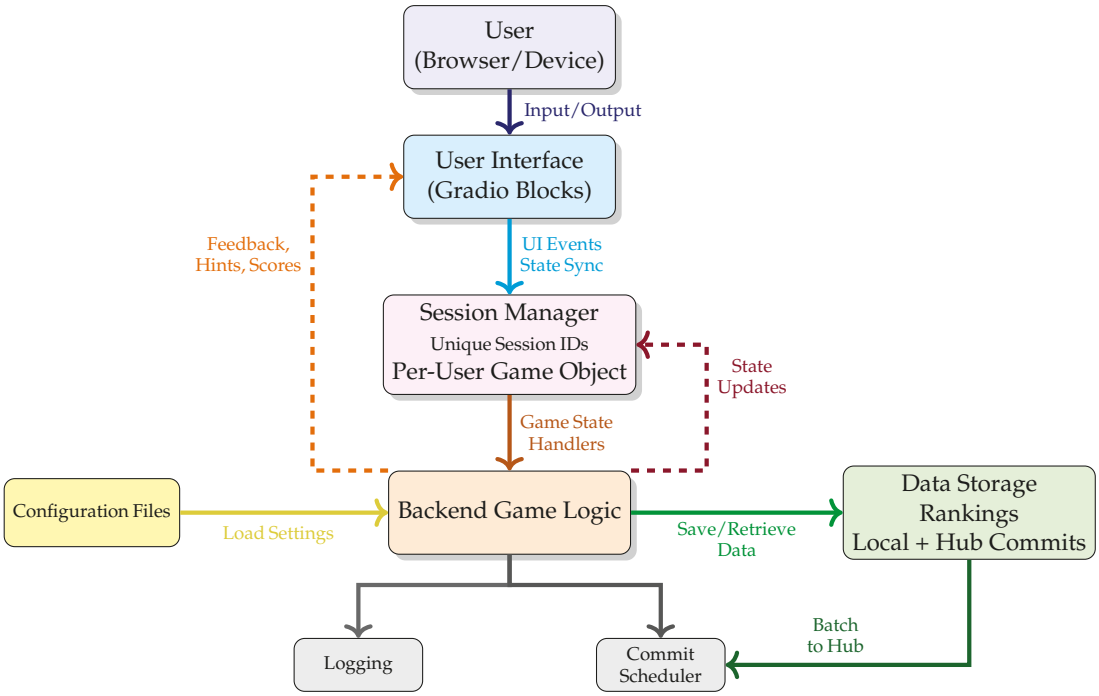


Figure 4. Overall software architecture of the Semantrix web application, illustrating user interface components, session isolation, backend/game logic, configuration loading, data storage, commit handling, and logging. Arrows indicate interaction and data flow; colored regions indicate major design concerns.

5.1. Application Architecture and Technology Stack

The Semantrix web application was implemented in Python (version 3.11.5) and leverages the Gradio Blocks API (version 5.21.0) to construct flexible, component-based user interfaces. The backend consists of custom Python classes for all game logic, state management, and interaction with natural language processing models.

The overall architecture and the relationships among core components are depicted in Figure 4. The main architectural layers are as follows:

- User Interface (Gradio Blocks): Handles all interactive elements (textboxes, radio buttons, action buttons, tables, and dynamic images) and user inputs/outputs.
- Session Manager: Maintains unique session identifiers and manages per-user game state isolation throughout each play session.

- Backend Game Logic: Implements core game mechanics, including game flow, input evaluation, scoring, hint generation, and state transitions.
- Configuration Files: JSON-based files used to load experimental parameters, user language, menu content, and model selection at runtime. Language is handled entirely via JSON. Per-session settings bind the UI language, the corresponding rules text, and the language-specific secret-word list. Hints are generated in the selected language via prompt templates. To change languages, only the configuration needs to be updated; the backend game logic remains unchanged.
- Data Storage (Rankings, Local + Hub Commits): Stores play sessions and rankings in local JSON files with automated saving and synchronisation to a Hugging Face Hub repository for persistent data management.
- Commit Scheduler: Batches and orchestrates data commits from local storage to the Hub for research data collection.
- Logging: Uses the Python logging module across all components for monitoring, debugging, and ensuring traceability and reproducibility.

5.2. User Interface and Experience

The user interface (UI) is dynamically constructed based on the game state, experimental condition, and user actions. Major UI components include the following:

- Header Markdown: Game instructions, experiment information, and session feedback.
- Image Display: Game logo and animated win/lose images for engagement.
- Textboxes and Buttons: For submitting guesses, requesting hints, and controlling game flow.
- Radio Buttons: For option selection (e.g., difficulty, or rules viewing).
- Ranking and History Tables: Dynamically updated tables show prior guesses, hint status, and user ranking.
- Session Timer: Ensures sessions are closed adequately on user inactivity, supporting user experience and experimental cleanliness.

Stateful interactions are managed via Gradio's `State` and `BrowserState` objects, with event-driven updates coordinated by backend functions.

5.3. Session Management and Request Handling

The Semantrix web application is deployed on Hugging Face Spaces, enabling global, concurrent user participation. Each user session is fully isolated to guarantee experimental integrity and fairness when evaluating user interactions with language models. Every user is assigned a unique session identifier and experiences a distinct game instance. This prevents participant interference and ensures that user inputs, feedback, and performance data are recorded independently.

While session isolation is a standard engineering practice in web applications, in this context, it is essential for providing reliable, reproducible data for analysing gameplay and user interactions.

This approach not only preserves data integrity but also creates a controlled environment for investigating the effectiveness and behaviour of adaptive hint generation and language understanding in real-time gameplay scenarios.

5.4. Configuration Files and Experimental Setup

As shown in Figure 4, configuration files are loaded by the Backend Game Logic at application start. Before any user session, these files specify the user's language, experimental parameters, and model selection. While it is possible to modify these files and update conditions between sessions, configuration changes do not take effect within an

active game; adjustments are applied only when the application (or session) is re-initialised. This design allows the following features:

- Switching user language and updating the corresponding UI, before the start of a session.
- Selecting embedding models (e.g., Sentence Transformers or Word2Vec) and controlling hint availability to support different experimental conditions, without modifying the core code, but only at start time.
- Flexibly adjusting or extending experiments by editing the condition JSON files before launching a new session.

5.5. Data Storage, Logging, and Commit Operations

All gameplay data are stored in local JSON files, segregated by session ID, for efficient tracking and easy export. Rankings and play histories are also session-bound. Periodically, the system batches and commits local user data to a Hugging Face Hub repository. This automated process enables collaborative research, reproducibility, and centralised analytics, while cleaning up local files after successful commits to optimise storage.

5.6. Extensibility and Customization

Semantrix's modular architecture ensures that new languages, embedding models, or experimental conditions can be incorporated with minimal code changes, usually only requiring updates to configuration files or model wrappers. UI elements can be rapidly adapted thanks to the flexibility of the Gradio Blocks API, and additional analytics or feedback modules can be integrated as required.

6. Methods

This section outlines the methodological approach followed to investigate the effects of semantic model selection and adaptive support on user experience in the Semantrix web application. Drawing from established guidelines in human–computer interaction research, our methods employed a preregistered, between-subject factorial design with random assignment to conditions, adherence to research ethics and GDPR compliance, and using validated self-report questionnaires alongside objective behavioural metrics. By deploying the study in a real-world, web-based setting while systematically controlling experimental variables, we sought to maximise both robustness and ecological validity.

6.1. Rationale and Overall Study Design

The design of Semantrix targets a central question in HCI and intelligent systems: How do underlying computational mechanisms and system-level scaffolds (such as adaptive hints) alter both the measurable and felt quality of human–digital game interaction? Rather than focusing solely on technical accuracy or raw performance, we aimed to develop a rich, multidimensional understanding of user experience, capturing engagement, perceived competence, enjoyment, and the subtle interplay between user intent and system feedback.

To this end, we adopted a 2×2 between-subject factorial design. Several considerations drove this choice. First, manipulating both the semantic embedding model (Sentence Transformers, representing a state-of-the-art contextual approach, and Word2Vec as a widely recognised baseline) and the presence or absence of dynamic hinting allow us to independently and jointly assess their contributions to end-user outcomes and system usability. To limit factorial growth and preserve power, we fixed the hint-trigger policy to the EMA-based detector described in Section 4.3. Second, the between-subject paradigm, wherein each participant experiences only one experimental condition, was preferred to within-subject alternatives to preclude confounding effects due to carry-over, learning,

or fatigue. This design minimises the risk that experiences in one condition would bias or influence responses in another, an important consideration when eliciting sensitive subjective appraisals and minimising participant burden [19]. Before data collection, all aspects of the study protocol, including hypotheses, variable manipulations, and planned analyses, were preregistered. This step affirmed the transparency and reproducibility of our experiment, aligning our approach with ongoing recommendations in computational and behavioural research communities [20].

All participants provided informed consent, and data handling procedures complied with the General Data Protection Regulation (GDPR, Regulation (EU) 2016/679). (The data processing activity procedure is recorded in the public University Processing Activity Log (https://docs.google.com/document/d/1ztSm0rRHiY5gCQOk7s3mfsBcX9x5xD2qQ9C5Go0AHDk/edit?usp=sharing. accesed on 12 February 2025)

6.2. Experimental Conditions and Interventions

At the heart of our experiment were two independent variables, each modelled as a binary factor:

1. Semantic Embedding Model. Participants either interacted with a version of the game powered by a Sentence Transformer model capable of nuanced and context-sensitive semantic similarity judgments or a more traditional Word2Vec model, which relies on static word-level vector relationships. The rationale for this contrast is twofold: While Sentence Transformers have demonstrated superiority in capturing context and meaning in NLP research, it remains an open empirical question whether such technological advances meaningfully translate into richer, more satisfying user experiences in a live system context. Moreover, direct comparison allows us to contextualise the added value of state-of-the-art AI within the lived realities of end-users.
2. Dynamic Hint System. We further established whether participants received only basic semantic score feedback or whether the full dynamic hint system was enabled, providing progressively more informative cues and support when users encountered difficulty or stagnation. This adaptive support mechanism is intended to enhance perceived competence and enjoyment by offering timely help, though it may also reduce perceived challenge or the sense of autonomy when overused, a trade-off noted in prior educational research [21]; thus, its effects are subject to empirical examination.

This resulted in four distinct conditions:

- (1) Sentence Transformer + Dynamic Hints: Users experience both advanced semantic scoring and adaptive feedback.
- (2) Sentence Transformer + No Dynamic Hints: Advanced scoring, minimal feedback.
- (3) Word2Vec + Dynamic Hints: Classic embedding paired with adaptive support.
- (4) Word2Vec + No Dynamic Hints: Baseline in both model and feedback.

Random assignment to these conditions was implemented automatically upon each participant's entry to the web application, ensuring balanced condition sizes and controlling for self-selection and diurnal or batch effects. Figure 5 provides an overview of the participant flow, from entry and random assignment to session activities and survey invitation. All phases were fully automated in the backend.

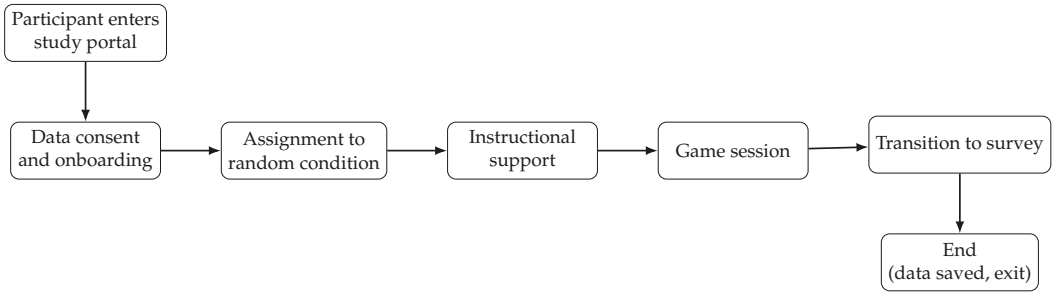


Figure 5. Experimental procedure flow for participant sessions in the Semantrix study. All steps, including random assignment, session management, and data saving, are automated in the backend.

6.3. Participants and Recruitment

Our sampling strategy prioritised diversity and external validity while remaining feasible for a controlled experimental deployment. We recruited 42 individuals (age: 18–64, $M = 30.1$ years), using multimodal online outreach (university mailing lists, professional association newsletters, public social media groups). This approach was chosen to avoid over-reliance on a single, possibly homogeneous population and reflect contemporary digital platforms' broad demographic reach.

Of the 42 participants, 31 (73.81%) identified as women, 10 (23.81%) as men, and 1 (2.38%) preferred not to disclose their gender. The educational level was high: 81% (34) had or were pursuing a university degree (bachelor's or higher), and 15 reported having a master's or PhD. Three reported completing only secondary education (high school or vocational training). Participants also varied in their self-assessed familiarity with technology: 14 rated themselves as "high" familiarity, 7 as "medium-high", 15 as "medium", 3 as "medium-low", and 3 as "low". Overall, the sample skewed toward higher formal education and moderate to high digital literacy. While this may align with the potential primary users of web-based language games, it may limit the generalisability of our findings to populations with lower education levels or less experience with digital games and applications based on NLP.

All participants reported normal or corrected-to-normal vision, which was required for engagement with the visual interface and fair comparison of usability metrics. After reviewing a detailed privacy and data protection statement (following GDPR), each prospective participant provided informed, digitally recorded consent. Completing the post-game survey and open-ended comment fields was entirely optional to ensure voluntary participation and minimise pressure, with no compensation contingent on these elements.

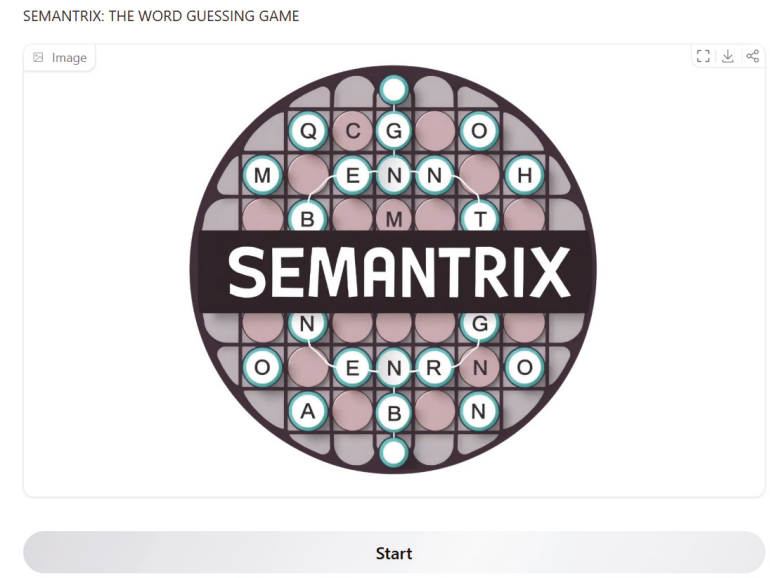
6.4. Experimental Procedure

The entire participant experience was designed to be seamless and unobtrusive, mirroring naturalistic interaction with web-based entertainment applications. The experimental flow is summarised in Figure 5. The UI sequence that instantiates this flow is shown in Figure 6, which maps onto the game phases in Figure 2: initialisation (green; A1a–e), main loop (blue; A1f–g), and game end (pink; A1h).

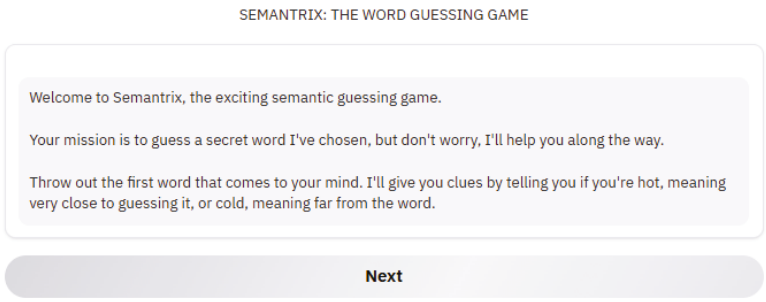
1. Data Consent and Onboarding: After reading the data protection summary and providing informed consent, participants reached the landing and welcome screens (Figure 6a,b), which introduced the goal and the "hot/cold" feedback idea. This corresponds to the initialisation phase in Figure 2.
2. Instructional Support: Participants were asked whether they wished to view the rules (Figure 6c). The rules and scoring overview (semantic score 0–10 and best-tries ranking) were then presented (Figure 6d), followed by the ready-to-start screen before the first guess (Figure 6e). These steps complete the initialisation phase in Figure 2.

3. Game Session: Participants played one or more rounds, guessing the secret word with semantic feedback. The gameplay view (Figure 6f) showed the best-tries table and last-guess feedback. When the EMA-based detector (Figure 3) indicated participants getting stuck, the system displayed a dynamic hint panel (Figure 6g; Word Ranking shown as an example). These elements instantiate the main loop in Figure 2.
4. Transition to Survey: At the end of a round, the interface displayed the solution and a short "curiosity" fact, with options to play again or finish (Figure 6h), after which participants were invited (optional) to complete the post-game survey. This corresponds to the game-end phase in Figure 2.

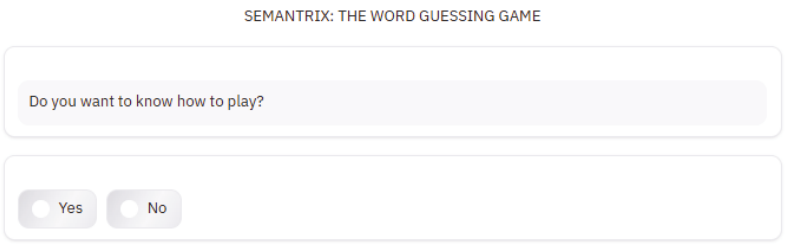
All experimental operations, from randomisation to data saving, were automated in the backend to minimise demand effects or unintentional bias.



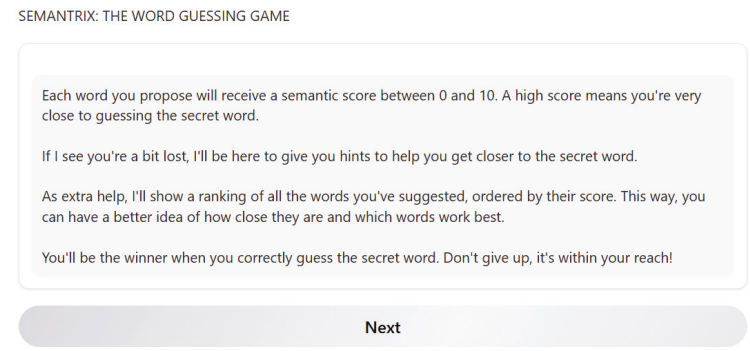
(**a**)



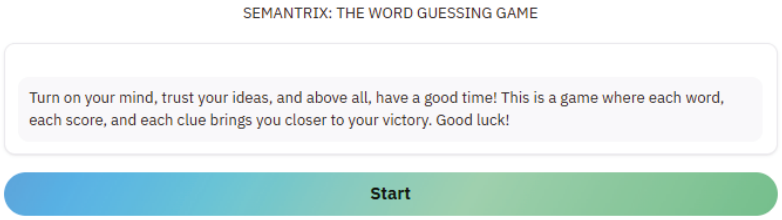
(**b**)



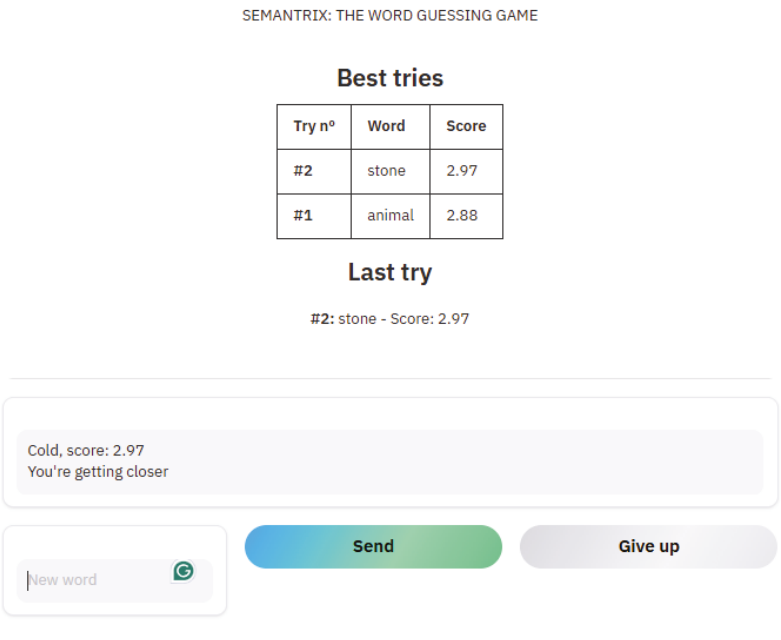
(**c**)


SEMANTRIX: THE WORD GUESSING GAME
Each word you propose will receive a semantic score between 0 and 10. A high score means you're very close to guessing the secret word.
If I see you're a bit lost, I'll be here to give you hints to help you get closer to the secret word.
As extra help, I'll show a ranking of all the words you've suggested, ordered by their score. This way, you can have a better idea of how close they are and which words work best.
You'll be the winner when you correctly guess the secret word. Don't give up, it's within your reach!
Next

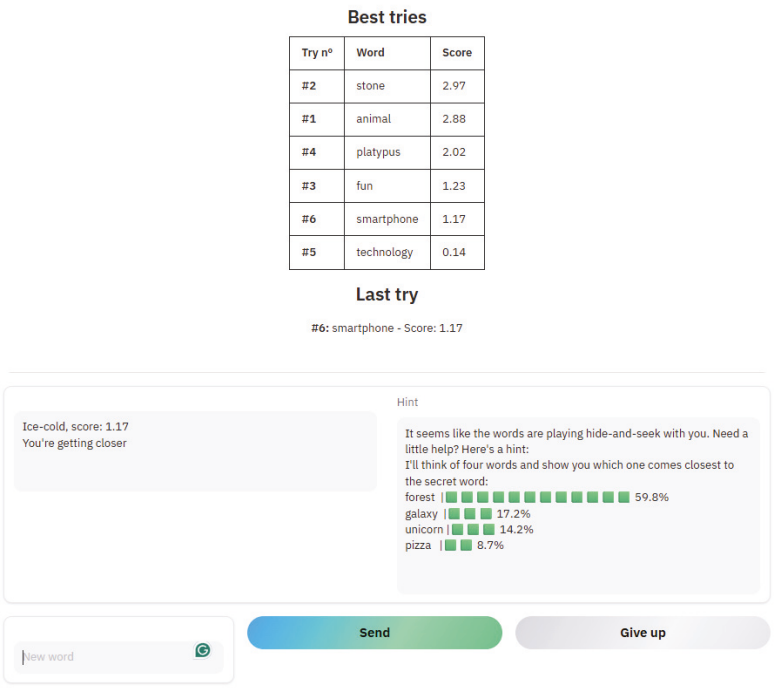
(**d**)


SEMANTRIX: THE WORD GUESSING GAME
Turn on your mind, trust your ideas, and above all, have a good time! This is a game where each word, each score, and each clue brings you closer to your victory. Good luck!
Start

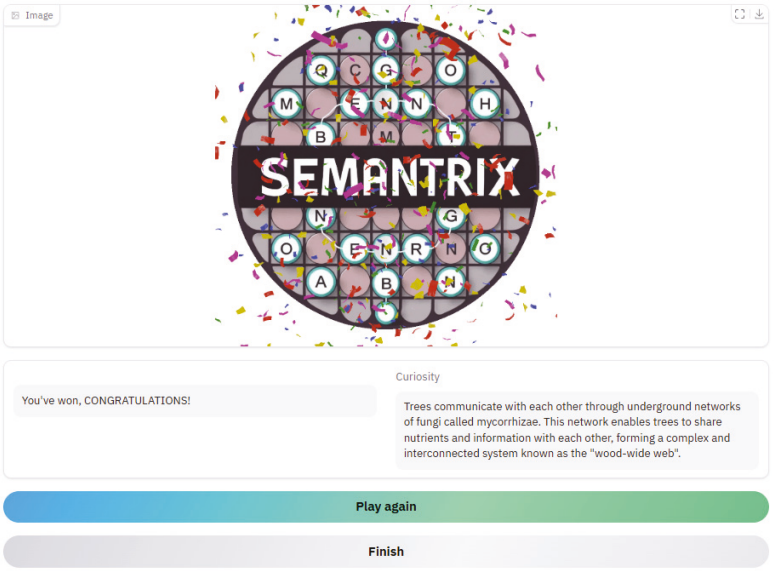
(**e**)


SEMANTRIX: THE WORD GUESSING GAME
Best tries
Try nº
Word
Score
#2
stone
2.97
#1
animal
2.88
Last try
#2: stone - Score: 2.97
Cold, score: 2.97
You're getting closer
New word
Send
Give up

(**f**)



(**g**)



(**h**)

Figure 6. Semantrix participant interface, from onboarding to completion. Screens illustrate the full flow: landing, welcome and rules, game start, live gameplay with semantic feedback, adaptive hinting, and end-of-round summary. (**a**) Landing page with logo and "Start" call to action, (**b**) welcome message introducing the goal and feedback, (**c**) prompt asking whether to view the rules (yes/no), (**d**) rules and scoring overview (semantic score 0–10; best-tries ranking), (**e**) ready-to-start screen before the first guess, (**f**) gameplay view with best-tries table and feedback for the last guess, (**g**) Dynamic hint panel (Word Ranking; initial-tier hint showing relative proximity), (**h**) end-of-round screen with solution, "curiosity" fact, and play-again options.

6.5. Measurement Instruments and Rationale

Game Metrics: Throughout gameplay, fine-grained behavioural data were collected, leveraging the affordances of digital logging in web applications. Logged metrics encompassed the following:

- Number of rounds played: Serves as a broad proxy for engagement and perseverance.
- Attempts and correct guesses: Support learning and error patterns analysis.
- Cumulative/average semantic scores: Permit quantification of performance quality under different feedback regimes.
- Hint requests and hint usage: Directly inform the relationship between support structure and observed reliance on system scaffolding.
- Completion times: Offer additional insight into cognitive load, fluency, and decision strategies.

These measures offer the potential to disentangle qualitative phenomena such as engagement and challenge from their behavioural correlates.

Self-Reported Metrics: To capture the multi-faceted and often subtle subjective experience of interacting with Semantrix, we assembled a unified post-experiment questionnaire incorporating validated constructs from contemporary HCI and psychology. Questionnaire selection was informed by coverage of relevant theoretical domains, prior use in engagement and digital entertainment studies, and brevity to minimise user fatigue. Components consisted of the following:

- User Engagement Scale—Short Form (UES-SF) [22]: This questionnaire measures four dimensions: Focused Attention, Perceived Usability (reverse-coded), Aesthetic Appeal, and Reward Factor. We included all four subscales in our study, using the full set of 12 questions.
- Intrinsic Motivation Inventory (IMI) [23]: From this questionnaire, we administered only the Interest/Enjoyment and Perceived Competence subscales, well established as being predictive of engagement and positive appraisal in digital gameplay. A total of 13 items from these subscales were used.
- FunQ [24]: We used this questionnaire by selecting the dimensions most relevant to our research focus: immersion, delight, challenge, and stress. These dimensions were chosen to align with the specific aspects of user experience we aimed to investigate in the context of Semantrix. The Loss of Social Barriers and Autonomy dimensions were omitted, as they were less relevant to our study objectives and the solitary nature of the game. In total, 12 items from this adapted version were used.

In total, participants were presented with a fixed sequence of 33 questions. The order of the questions was not randomised, ensuring that all participants shared the same experience and facilitating cross-user comparison. We took active steps to minimise redundancy by cross-referencing items for conceptual overlap during questionnaire integration. For non-English-speaking participants, a rigorous translation and back-translation protocol (following ITC guidelines) was enacted to preserve each item's semantic and psychometric integrity.

Scoring: All survey items used a consistent five-point Likert scale (1 = strongly disagree, 5 = strongly agree). For analytic consistency, scores on each construct (e.g., engagement, motivation) were computed as the arithmetic mean of their respective items, except for the FunQ, where a sum score approach, reflecting factor structure, was used, as recommended by the original authors. Open-ended feedback provided qualitative insights, supplementing quantitative results.

6.6. Hypotheses

Grounded in theoretical models of user experience in digital games and prior findings on adaptive system support and intelligent interfaces, we formulated distinct sets of hypotheses for three core experiential constructs: engagement, intrinsic motivation, and fun. Each construct was operationalised through validated measurement questionnaires as described above.

Engagement, comprising attentional focus, aesthetic appreciation, ease of use, and willingness to re-engage, is a key predictor of positive user experiences in digital interaction [22]. Building on existing literature suggesting that both system intelligence and adaptive feedback can play roles in sustaining user attention and immersion, we hypothesised the following:

- H_{eng1}: There will be significant differences in reported engagement as a function of the underlying semantic model; specifically, participants experiencing the Sentence Transformer-based system will report higher levels of engagement than those assigned to the Word2Vec-based system.
- H_{eng2}: Activation of the dynamic hint system will yield higher self-reported engagement compared to the minimal-feedback conditions.
- H_{eng3}: There will be significant interaction effects such that the combination of a sophisticated semantic model and rich dynamic feedback will produce higher engagement scores than can be attributed to either variable in isolation.

Intrinsic motivation reflects the degree to which users perceive the activity as inherently interesting, enjoyable, and satisfying, independent of external rewards, a critical driver of persistence and deep learning in serious games and HCI contexts [23]. Based on self-determination theory and prior empirical work with adaptive systems, we proposed the following hypotheses:

- H_{im1}: The use of the Sentence Transformer model will result in significantly higher intrinsic motivation scores relative to the Word2Vec model, as advanced semantic understanding may promote curiosity and perceived task meaningfulness.
- fH_{im2}: Participants with access to the dynamic hint system will report greater intrinsic motivation, reflecting enhanced feelings of competence and effective challenge.
- fH_{im3}: The impact of dynamic hints on intrinsic motivation will be amplified in the context of the advanced semantic model, resulting in a statistically significant interaction effect.

Fun encapsulates both affective enjoyment and playful challenge, underpinning the appeal of many serious games [24]. It was conceptualised here not merely as hedonic pleasure but as a multidimensional construct comprising immersion, delight, and the balance of challenge and stress:

- H_{fun1}: Participants in the Sentence Transformer conditions will report higher levels of overall fun, as advanced semantic scoring is expected to create a more dynamic and responsive interaction, promoting states of flow and playful absorption.
- H_{fun2}: Activation of the dynamic hint system will be associated with increased fun ratings, by supporting playful exploration while buffering against excessive frustration.
- H_{fun3}: The presence of both an advanced semantic model and a dynamic hint system will have a synergistic effect, yielding the highest observed fun scores and demonstrating a significant interaction between the two factors.

Each set of hypotheses were tested using statistical analyses selected by the observed properties of the collected data. When assumptions such as multivariate normality and adequate correlation among the dependent variables were satisfied, factorial MANOVA was applied to investigate the main and interaction effects of system intelligence and feedback. The final analytical approach was determined after verifying the suitability of these statistical assumptions.

7. Results

This section reports the results of the Semantrix online platform experiment, including in-game behavioural data and self-reported user experience measures. We first present the

core user outcome metrics across experimental conditions, followed by formal statistical analyses addressing our preregistered hypotheses ($H_{eng1,2,3}$, $H_{im1,2,3}$, $H_{fun1,2,3}$). The statistical analyses were carried out using the IBM SPSS Statistics software (version 26) and the information extracted from the comments left by the 42 participants of the experiment. An alpha significance level of $\alpha = 0.05$ was used for the statistical analyses.

7.1. User Outcome Metrics

To assess user engagement, task performance, and experience, we collected a range of behavioural (e.g., playing time, percentage of games won, hints received, word attempts, rounds completed) and self-report measures (UES–SF for engagement, FunQ for fun, IMI for intrinsic motivation). Tables 2 and 3 summarise these outcome variables by experimental condition.

Table 2. Game metrics by experimental condition. Reported as mean ± SD (games won shown as a percentage).

Model	**Sentence Transformers**	**Sentence Transformers**	**Word2Vec**	**Word2Vec**
Hints	**Enabled**	**Disabled**	**Enabled**	**Disabled**
Playing time (sec/round)	169.8 ± 147.3	457.8 ± 443.5	358.8 ± 363.9	101.8 ± 117.4
Games won (%)	58.3%	54.2%	53.3%	75.0%
Hints received per round	1.31 ± 1.25	-	3.00 ± 3.26	-
Word attempts per round	10.0 ± 7.8	27.0 ± 24.6	19.7 ±18.9	8.4 ± 8.2
Played rounds	2.00 ± 1.28	1.50 ± 0.67	1.75 ± 1.22	1.75 ± 0.50

Notably, substantial differences in gameplay behaviour were observed across conditions, as shown in Table 2. Conditions with Sentence Transformers and/or enabled hints were characterised by longer average playing times per round and more word attempts per round, particularly notable in the Sentence Transformers + Hints group (mean playing time 169.8 ± 147.3 s/round, mean word attempts 10.0 ± 7.8), compared to the much shorter duration in the Word2Vec + No Hints condition (101.8 ± 117.4 s/round, mean word attempts 8.4 ± 8.2).

The number of rounds played per user was also highest in the Sentence Transformers with hints (2.00 ± 1.28), again suggesting that the combination of more advanced semantic processing and adaptive hinting fostered greater persistence and re-engagement.

These game metrics could indicate that advanced language models and dynamic hints contributed to greater user persistence (longer play sessions, more attempts, and rounds played). In contrast, their absence, most markedly in the Word2Vec + No Hints group, was associated with shorter sessions and reduced behavioural engagement. These objective patterns reinforce the trends observed in the self-report engagement, fun, and motivation scores.

Longer playing times and more hints received were thus most prominent when dynamic hints were available, especially in the Sentence Transformers + Hints group. This convergence of game and self-report data could suggest that adaptive feedback and the use of advanced semantic models both sustain user engagement and promote persistence in solving the game.

Descriptively, the self-reported outcomes in Table 3 show a coherent pattern across conditions. The combined condition (Sentence Transformers + Hints) reached the highest means for engagement (UES-SF = 3.63 ± 0.46; 95% CI: 3.33–3.92), fun (FunQ = 3.87 ± 0.25; 95% CI: 3.71–4.03), and intrinsic motivation (IMI = 3.86 ± 0.53; 95% CI: 3.52–4.20). Disabling the hint system was associated with lower ratings under both models, with the largest declines observed for IMI: from 3.86 to 2.67 with Sentence Transformers and from 3.30 to 2.71 with Word2Vec. When hints were available, Sentence Transformers slightly

exceeded Word2Vec on UES-SF (3.63 vs. 3.56) and FunQ (3.87 vs. 3.79), whereas motivation appeared primarily driven by hint availability rather than model choice. The weakest profile corresponded to Word2Vec without hints (UES-SF = 3.15; FunQ = 3.32; IMI = 2.71), though the wider confidence intervals in this group, reflecting its smaller sample size (n = 5; see Section 7.3), suggest caution. These descriptive trends motivate the inferential analyses reported next. While both game metrics and self-reported questionnaire data were collected as outcome measures, our hypothesis testing and subsequent multivariate analyses focused specifically on the questionnaire-based measures: UES-SF (engagement), FunQ (fun), and IMI (intrinsic motivation).

Table 3. Questionnaire score metrics by experimental condition. Reported as mean ± SD [95% CI]. Questionnaire scores (UES-SF, FunQ, IMI): a higher value indicates more positive self-reported outcomes; bold marks the highest mean within each row (bolding does not imply statistical significance).

Model	**Sentence Transformers**		**Sentence Transformers**		**Word2Vec**		**Word2Vec**	
Hints	**Enabled**		**Disabled**		**Enabled**		**Disabled**	
UES–SF (Engagement)	**3.63 ± 0.46**	**[3.33, 3.92]**	3.17 ± 0.60	[2.79, 3.55]	3.56 ± 0.32	[3.35, 3.76]	3.15 ± 0.86	[2.08, 4.22]
FunQ (Fun)	**3.87 ± 0.25**	**[3.71, 4.03]**	3.22 ± 0.61	[2.83, 3.60]	3.79 ± 0.36	[3.56, 4.01]	3.32 ± 0.54	[2.65, 3.99]
IMI (Motivation)	**3.86 ± 0.53**	**[3.52, 4.20]**	2.67 ± 0.58	[2.30, 3.04]	3.30 ± 0.56	[2.94, 3.65]	2.71 ± 0.69	[1.85, 3.56]

7.2. Assumption Checks

Before conducting multivariate analysis, we assessed all primary outcome variables for the relevant assumptions. Shapiro–Wilk tests indicated no significant deviations from normality for UES–SF ($p = 0.087$), FunQ ($p = 0.078$), or IMI ($p = 0.539$). Levene's tests also indicated that the assumption of homogeneity of variances was satisfied (all $p > 0.05$). Furthermore, inspection of Pearson correlation coefficients among the three dependent variables showed moderate and statistically significant positive correlations: UES–SF and FunQ ($r = 0.36$, $p = 0.022$), FunQ and IMI ($r = 0.76$, $p < 0.001$). The correlation between UES–SF and IMI was weaker and not significant ($r = 0.19$, $p = 0.236$). These results indicate that, while the constructs are related, they are not redundant, and the intercorrelation is sufficient to justify a multivariate analysis approach. Given that all necessary assumptions were satisfied, we proceeded with a 2×2 factorial MANOVA to evaluate the effects of the semantic embedding model and the dynamic hint system on the outcome measures.

7.3. Participant Decay and Condition-Specific Dropout

During data collection, a pronounced pattern of participant dropout was observed in one experimental condition: Word2Vec + Hints Disabled. Out of 15 users who entered this group, only 5 completed the post-game self-report questionnaire. In contrast, completion rates were substantially higher in all other conditions (Sentence Transformer + Hints, Sentence Transformer + No Hints, Word2Vec + Hints), with 12 valid responses per condition.

This markedly higher decay rate (only 33.3% completion in Word2Vec + No Hints compared to nearly 85% in other conditions) is noteworthy. Most participants either exited the game before reaching the survey or chose not to continue after the gameplay session, suggesting a significant lack of engagement or dissatisfaction uniquely associated with that configuration. This outcome could indicate reduced engagement, precisely the facet measured by our primary dependent variables, and aligns with our finding that advanced models and/or hinting systems are critical for sustained user involvement in a semantic guessing game context.

Consequently, for subsequent statistical analyses, the condition with Word2Vec Model + No Hints contained a much reduced sample size ($n = 5$), while other conditions each included $n = 12$ participants. This imbalance was taken into account in the interpretation of inferential statistics (see Section 8). Nonetheless, this dropout phenomenon

may reflect reduced engagement without semantic modelling and adaptive support, although we cannot definitively attribute causality to these factors.

7.4. Statistical Analysis and Hypothesis Testing

A 2 × 2 multivariate analysis of variance (MANOVA) was conducted to examine the main and interaction effects of semantic embedding model (Sentence Transformers vs. Word2Vec) and Dynamic Hint System (enabled vs. disabled) on engagement (UES–SF), perceived fun (FunQ), and intrinsic motivation (IMI). This approach directly addressed our primary preregistered hypotheses ($H_{eng,1,2,3}$, $H_{fun,1,2,3}$, $H_{im,1,2,3}$).

The main effect of the Model factor ($Wilk's\Lambda = 0.696, F(4,42) = 4.484, p = 0.004$, $\eta^2 = 0.304$) indicated that, across all conditions, participants using Sentence Transformers reported higher engagement and fun than those using Word2Vec (cf. Table 4, Figure 5). The main effect of the Hints factor ($Wilk's\Lambda = 0.544, F(4,42) = 8.585, p < 0.001, \eta^2 = 0.456$) showed that, independent of Model, enabling dynamic hints significantly increased all three outcomes (engagement, fun, and motivation), supporting hypotheses H_{eng2}, H_{fun2}, and H_{im2}. The main effect for Model was not significant for intrinsic motivation ($p = 0.075$), thus only partially supporting H_{im1}.

Crucially, a significant *interaction effect* (Model × Hints, $Wilk's\Lambda = 0.728$, $F(4,42) = 3.835, p = 0.010$, $\eta^2 = 0.272$) demonstrates that the effect of one variable depends on the level of the other. For instance, as detailed in Table 4, the benefit of dynamic hints on engagement and fun was especially strong when combined with the Sentence Transformer model. For motivation (IMI), the combination produced a larger increase than either factor alone. Still, the motivational benefit of Sentence Transformers became apparent mainly when hints were absent, suggesting possible "redundancy" in their combined application (see Figure 7).

To clarify individual variable effects concerning our hypotheses, we conducted Bonferroni-corrected univariate ANOVAs for each dependent variable. Table 4 details the test statistics for all main and interaction effects. The estimated marginal means (±95% CI) for engagement (UES-SF), fun (FunQ), and intrinsic motivation (IMI) by Model and Hint condition are shown in Figures 7–9. These plots complement the main and interaction effects reported in Table 4.

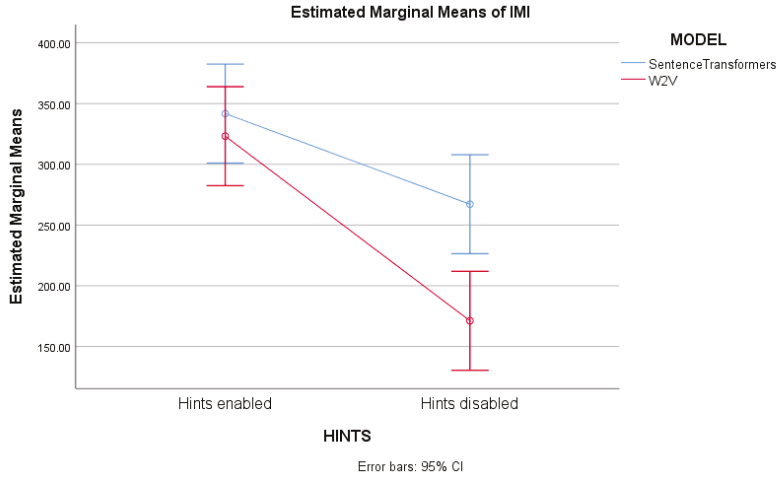


Figure 7. Estimated marginal means (±95% CI) of IMI (intrinsic motivation) by Model and Hint condition in the 2 × 2 design. Note: Periods indicate decimal points.

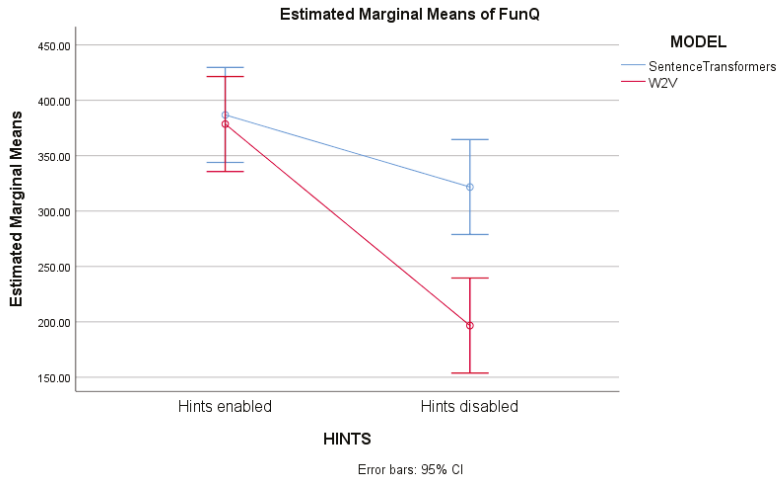


Figure 8. Estimated marginal means (±95% CI) of FunQ (fun) by Model and Hint condition in the 2 × 2 design. Note: Periods indicate decimal points.

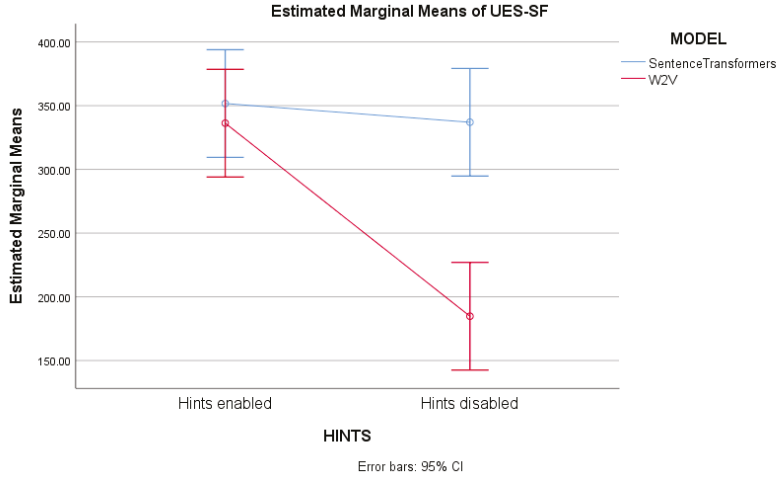


Figure 9. Estimated marginal means (±95% CI) of UES-SF (engagement) by Model and Hint condition in the 2 × 2 design. Lines connect hint conditions within each model; higher values indicate a more positive self-report. Note: Periods indicate decimal points.

Table 4. Bonferroni-corrected two-way ANOVAs for each dependent variable. Statistics and significance levels are shown for the main effects (Model, Hints) and their interaction. Rightmost column lists hypotheses supported by each effect ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$).

		Main Effects		Interaction		Partial η^2	Supported
		$F(1, 45)$	p	$F(1, 45)$	p		Hypotheses
UES–SF	Model	13.34	0.001 **			0.233	H_{eng1}
	Hints	13.08	0.001**			0.229	H_{eng2}
	Interaction			13.79	0.001 **	0,239	H_{eng3}
FunQ	Model	9.81	0.003 **			0.182	H_{fun1}
	Hints	33.67	<0.001 ***			0.434	H_{fun2}
	Interaction			7.53	0.009 *	0,146	H_{fun3}
IMI	Model	3.33	0.075			0.07	—
	Hints	23.85	<0.001 ***			0.352	H_{im2}
	Interaction			8.22	0.006 **	0,157	H_{im3}

Across outcomes found in Table 4, dynamic hints showed large effects, $\eta_p^2 = 0.229$ on engagement (UES-SF), 0.434 on fun (FunQ), and 0.352 on intrinsic motivation (IMI). The semantic model showed moderate-to-large effects on engagement ($\eta_p^2 = 0.233$) and fun

($\eta_p^2 = 0.182$), and a small-to-moderate effect on motivation ($\eta_p^2 = 0.070$). Interactions were moderate ($\eta_p^2 = 0.146 - 0.239$), indicating that Sentence Transformers amplified the benefit of hints.

In summary, engagement and fun benefited from both advanced models and dynamic hints, especially in combination, while adaptive hints most reliably enhanced motivation. This pattern supports all engagement and fun hypotheses and those relating to the motivational effects of adaptive support.

8. Discussion

This study aimed to elucidate how advanced semantic modelling and including a dynamic hint system influence engagement, intrinsic motivation, and fun in users interacting with an online semantic word-guessing game. Through a controlled factorial design, we were able to isolate the effects of Transformer-based semantic embeddings and adaptive feedback mechanisms on both behavioural and subjective metrics.

The results revealed clear patterns across the primary dependent variables: First, using Transformer-based embeddings and dynamic hints significantly enhanced user engagement as measured by the UES-SF. The effect was statistically robust and reflected behavioural persistence, with longer session times and higher completion rates in these conditions. The fact that engagement was most pronounced when both factors were combined highlights the importance of aligning technical capabilities (deep semantic matching) with the adaptivity of user support mechanisms.

User-reported fun, as assessed by the FunQ questionnaire, showed a parallel pattern. Both main effects and their interaction reached significance, with the significant fun reported under the dual presence of rich semantic modelling and dynamic hints. This is consistent with the theoretical view that enjoyment in digital games is fostered when the challenge is well matched to user abilities and is scaffolded in ways that avoid excessive stagnation or frustration.

Intrinsic motivation, measured via the adapted IMI, was reliably increased by providing adaptive hints but not as a main effect of the underlying semantic model. Notably, a significant interaction was found, indicating that the motivational advantage of semantic Transformers manifests most strongly when hints are absent. This pattern suggests that motivational benefits may arise from either deeper semantic feedback or a supportive hint system, but not necessarily from both simultaneously, pointing to a kind of "motivational redundancy" in their combined application. Although the Model's main effect on intrinsic motivation did not reach significance ($p = 0.075$), the partial eta-squared ($\eta_p^2 = 0.07$) suggested a small-to-moderate trend. Together with the significant Model × Hints interaction and the estimated marginal means, shown in Figure 7, this is compatible with a motivational redundancy theory: adaptive hints robustly increase perceived competence and thus intrinsic motivation. At the same time, richer semantic feedback from Sentence Transformers contributes to motivation mainly when hints are disabled. On the other hand, we note that the unequal sample sizes, particularly in the Word2Vec + No Hints condition, may have reduced sensitivity to small main effects on IMI; estimates should therefore be treated as relative (see Section 8 for details). In terms of use cases, we can say that these results could point to the importance of detecting the moment and adjusting the appropriate level of hints to preserve the "productive struggle" and, on the other hand, suggests that advanced semantic modelling may be particularly valuable for maintaining motivation in contexts where the hint system is unable to provide sufficient support to the user, while more consistently benefiting engagement and enjoyment in all circumstances.

The observed effects may be understood through cognitive and affective mechanisms. Enhanced semantic modelling likely enables more accurate and rewarding feedback; users

can sense that the system "understands" their intent at a meaning-based level, which can foster a sense of competence and relatedness. Meanwhile, the adaptive hint system prevents stagnation, averts frustration, and may restore user self-efficacy when progress stalls. Both systems' synergetic effect maintains a "productive struggle" that is optimal for engagement and positively affects gameplay. The effect sizes obtained, particularly for dynamic hints, suggest practical significance beyond statistical reliability, while the small-to-moderate model effect on motivation ($\eta_p^2 = 0.07$) indicates a trend that may be resolved with larger samples.

Although we adopted an EMA-based trigger, other policies are reasonable as well. Rule-based triggers (e.g., triggering after a fixed number of failed attempts, or when scores remain below a threshold) are simple and transparent. Still, they may ignore the magnitude and direction of change and require frequent readjustments, sometimes triggering after harmless oscillations. Time-based triggers are easy to implement, but they combine reflection time, writing speed, and device or network latency, so they may not reliably indicate that a player has been lost. On the other hand, user-initiated hints maximise autonomy and transparency but can introduce biases that complicate comparisons between conditions; in the pilot tests where we enabled the option to request hints, several participants avoided asking for help even when they got stuck, which increased the variability in frustration. In contrast, our EMA-based policy is lightweight and reacts to trends rather than noise or peaks. The two-stage EMA, described above, reduces fluctuation while maintaining sensitivity to avoid getting stuck. It still requires parameter choice and a brief warm-up period.

Limitations

While our results are promising, several limitations must be acknowledged. Though adequate for detecting main effects, the sample size was restricted in some subgroups due to pronounced participant dropout. Notably, in the condition lacking advanced semantics and dynamic hints (Word2Vec + No Hints), only a few participants completed the experiment. Such an imbalance can reduce power, widen confidence intervals, and may make interaction estimates more variable and sensitive to deviation from normality and equal variances across groups. The dropout in that condition may also be related to the outcomes (e.g., lower engagement), which could introduce bias into condition means. For example, if less-engaged participants were less likely to complete the survey, the remaining users might overestimate engagement. To minimise these risks, we checked distributional assumptions, reported effect sizes and confidence intervals, and used multivariate tests, which are commonly applied with unequal n's; even so, effects involving the smallest sample size, especially interactions, should be interpreted with caution.

Our group of participants was predominantly highly educated, and many reported having a medium to high level of technological knowledge. This may have several implications. First, it could produce ceiling effects in basic usability and understanding of instructions, which could overstate perceived ease of use. Secondly, digitally experienced users may be more tolerant of new patterns of interaction with the interface, as well as hints generated by the system, and may have different help-seeking behaviours than less experienced users. These factors could attenuate or amplify the effects observed, so external validity for populations with less digital experience should be considered provisional.

While advantageous for reach and ecological validity, the web-based study design can introduce selection biases related to digital access and user motivation, limiting the extent to which these results can be generalised across different user populations or deployment scenarios.

Finally, reliance on remote API calls for generative models (e.g., GPT-4) may constrain responsiveness and scalability for real-time or edge applications. Technical optimisations regarding cost, latency, and robustness should be addressed in future work.

9. Conclusions

This work demonstrated that integrating Transformer-based NLP models and adaptive hinting mechanisms significantly enhanced user engagement and satisfaction in web-based semantic guessing games. Our results showed that the combination of advanced semantic embedding (Sentence Transformers) and dynamic feedback yielded statistically significant improvements in self-reported engagement, fun, and motivation compared to static word embeddings and minimal feedback baselines. Notably, user attrition rates were dramatically reduced in adaptive conditions, supporting the necessity of both technological and interactional scaffolding to sustain meaningful human–AI interaction.

Beyond technical performance, our findings suggest potential implications for designing intelligent educational and entertainment systems, particularly in the context of semantic word games and similar adaptive interfaces. The successful deployment of Semantrix illustrates the value of responsive, personalised support in promoting richer and potentially longer-lasting user experiences within this domain.

While this study demonstrates the benefits of adaptive and semantic feedback for user engagement and motivation, several routes remain for further exploration that could broaden the implications and applicability of our findings. Increasing sample size and expanding demographic diversity would enable a more robust assessment of generalisability across varied populations and usage contexts. In addition, it would be helpful to adopt methodological control measures to reduce imbalance and improve statistical reliability, such as setting per-condition quotas based on a priori power analysis and using stratified randomisation to limit uneven distribution. To reinforce generalisation, it would be very beneficial to broaden the selection of participants and establish quotas by condition stratified by educational level and digital literacy. Alternative types of hints (e.g., iconographic and audio-supported) to better support non-gamers, older adults, and users with less digital literacy would be valuable.

Longitudinal studies may also help examine whether observed engagement and motivation improvements translate to lasting educational or cognitive benefits. Additionally, it would be beneficial to examine the effects of different feedback modalities (e.g., audio or visual cues) and variants in game structure (such as collaborative or competitive modes), to better understand the mechanisms underlying the observed effects and their potential transferability. Looking ahead, it might be useful to develop the alternatives mentioned (rule-based and time-based triggers, etc.) and explore a comparison that also considers user-initiated hints. It might also be insightful to examine a hybrid policy that combines EMA detection with user confirmation and a conservative timeout, along with sensitivity analyses of EMA parameters.

These directions are not exhaustive but suggest natural extensions of the present work. Addressing them would help establish a broader understanding of how adaptive feedback systems can be deployed effectively across settings, while clarifying the scope of their educational and motivational impact.

While preparing this work, the authors used GPT-4o, from OpenAI, and Grammarly, from Grammarly Inc., to improve the work's readability and language. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

Author Contributions: Conceptualization, J.S.-S., J.C.C.M. and Á.C.-G.; Methodology, J.S.-S.; Software, J.S.-S.; Validation, J.S.-S.; Formal analysis, J.S.-S. and Á.C.-G.; Investigation, J.S.-S.; Writing—original draft, J.S.-S.; Writing—review & editing, J.C.C.M.; Visualization, J.C.C.M.; Supervision, J.C.C.M., Á.C.-G. and M.A.S.; Project administration, M.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results has received funding from the following projects: Robots sociales para mitigar la soledad y el aislamiento en mayores (SOROLI), PID2021-123941OA-I00, funded by Agencia Estatal de Investigación (AEI), Spanish Ministerio de Ciencia e Innovación; and Robots sociales para reducir la brecha digital de las personas mayores (SoRoGap), TED2021-132079B-I00, funded by Agencia Estatal de Investigación (AEI), Spanish Ministerio de Ciencia e Innovación; and Portable Social Robot with High Level of Engagement (PoSoRo) PID2022-140345OB-I00 funded by MCIN/AEI/10.13039/501100-011033 and ERDF A way of making Europe.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yu, H.; Wen, J.; Zheng, Z. CAMeL: Cross-Modality Adaptive Meta-Learning for Text-Based Person Retrieval. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 4651–4663. [CrossRef]
2. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805. [CrossRef]
3. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.08774. [CrossRef]
4. Matuszek, C. Grounded Language Learning: Where Robotics and NLP Meet. In Proceedings of the Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 5687–5691. [CrossRef]
5. Onyebuchi, N.; Olusola, A.; Hamad, N.; Osawaru, B.; Adewusi, O. AI in education: A review of personalized learning and educational technology. *GSC Adv. Res. Rev.* **2024**, *18*, 261–271. [CrossRef]
6. Tan, D.Y.; Cheah, C.W. Developing a gamified AI-enabled online learning application to improve students' perception of university physics. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100032. [CrossRef]
7. Dalsgaard, T.S.; Schneider, O. A Unified Model for Haptic Experience. *ACM Trans. Comput.-Hum. Interact.* **2025**, *32*, 9 . [CrossRef]
8. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781. [CrossRef]
9. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014 ; Moschitti, A., Pang, B., Daelemans, W., Eds.; pp. 1532–1543. [CrossRef]
10. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv: 1907.11692.
11. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. MPNet: Masked and Permuted Pre-training for Language Understanding. *arXiv* **2020**, arXiv:2004.09297. [CrossRef]
12. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv* **2020**, arXiv:1911.02116. [CrossRef]
13. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084. [CrossRef]
14. Muennighoff, N.; Tazi, N.; Magne, L.; Reimers, N. MTEB: Massive Text Embedding Benchmark. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 2–4 May 2023; Vlachos, A.; Augenstein, I., Eds.; pp. 2014–2037. [CrossRef]
15. Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv* **2025**, arXiv:2507.06261. [CrossRef]
16. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D.d.L.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825. [CrossRef]

17. Lames, M.; McGarry, T. On the search for reliable performance indicators in game sports. *Int. J. Perform. Anal. Sport* **2007**, *7*, 62–79. [CrossRef]
18. Anggono, A.H. Investment Strategy Based on Exponential Moving Average and Count Back Line. *Rev. Integr. Bus. Econ. Res.* **2019**, *8*, 153–161.
19. Bethel, C.L.; Murphy, R.R. Review of Human Studies Methods in HRI and Recommendations. *Int. J. Soc. Robot.* **2010**, *2*, 347–359. [CrossRef]
20. Gunes, H.; Broz, F.; Crawford, C.S.; der Pütten, A.R.v.; Strait, M.; Riek, L. Reproducibility in Human-Robot Interaction: Furthering the Science of HRI. *Curr. Robot. Rep.* **2022**, *3*, 281–292. [CrossRef]
21. Sun, L.; Kangas, M.; Ruokamo, H.; Siklander, S. A systematic literature review of teacher scaffolding in game-based learning in primary education. *Educ. Res. Rev.* **2023**, *40*, 100546. [CrossRef]
22. O'Brien, H.L.; Cairns, P.; Hall, M. A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form. *Int. J. Hum.-Comput. Stud.* **2018**, *112*, 28–39. [CrossRef]
23. Deci, E.L.; Eghrari, H.; Patrick, B.C.; Leone, D.R. Facilitating internalization: The self-determination theory perspective. *J. Personal.* **1994**, *62*, 119–142. [CrossRef] [PubMed]
24. Tisza, G.; Markopoulos, P. FunQ: Measuring the fun experience of a learning activity with adolescents. *Curr. Psychol.* **2023**, *42*, 1936–1956. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

High-Density Neuromorphic Inference Platform (HDNIP) with 10 Million Neurons

Yue Zuo [1], Ning Ning [1], Ke Cao [2], Rui Zhang [2], Cheng Fu [1], Shengxin Wang [2], Liwei Meng [1], Ruichen Ma [1], Guanchao Qiao [1], Yang Liu [1] and Shaogang Hu [1,2,*]

[1] The State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu 610054, China; emvdyzuo@std.uestc.edu.cn (Y.Z.); ning_ning@uestc.edu.cn (N.N.); 2022340106004@std.uestc.edu.cn (C.F.); lw_meng@std.uestc.edu.cn (L.M.); ruichen.ma@std.uestc.edu.cn (R.M.); qcqiao@uestc.edu.cn (G.Q.); yliu1975@uestc.edu.cn (Y.L.)

[2] The Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China; 202322280215@std.uestc.edu.cn (K.C.); 202322280207@std.uestc.edu.cn (R.Z.); shengxin_wang@std.uestc.edu.cn (S.W.)

* Correspondence: sghu@uestc.edu.cn

Abstract: Modern neuromorphic processors exhibit neuron densities that are orders of magnitude lower than those of the biological cortex, hindering the deployment of large-scale spiking neural networks (SNNs) on single chips. To bridge this gap, we propose HDNIP, a 40 nm high-density neuromorphic inference platform with a density-first architecture. By eliminating area-intensive on-chip SRAM and using 1280 compact cores with a time-division multiplexing factor of up to 8192, HDNIP integrates 10 million neurons and 80 billion synapses within a 44.39 mm^2 synthesized area. This achieves an unprecedented neuron density of 225 k neurons/mm^2, over 100 times greater than prior art. The resulting bandwidth challenges are mitigated by a ReRAM-based near-memory computation strategy combined with input reuse, reducing off-chip data transfer by approximately 95%. Furthermore, adaptive TDM and dynamic core fusion ensure high hardware utilization across diverse network topologies. Emulator-based validation using large SNNs, demonstrates a throughput of 13 GSOP/s at a low power consumption of 146 mW. HDNIP establishes a scalable pathway towards single-chip, low-SWaP neuromorphic systems for complex edge intelligence applications.

Keywords: neuromorphic computing; high-density architecture; spiking neural networks; near-memory computing; ReRAM

1. Introduction

Neuromorphic computing, drawing inspiration from the remarkable efficiency of the biological brain, holds immense promise for developing low-power intelligent systems [1]. Significant achievements have progressively shaped the neuromorphic ecosystem [2–5]. However, realizing this promise for large-scale, bio-inspired spiking neural networks (SNNs) presents unique and formidable architectural demands.

Conventional high-performance architectures, such as GPUs and TPUs [6,7], while offering high theoretical performance through extensive parallel arithmetic units, are fundamentally mismatched with SNN characteristics. Their general-purpose nature is ill-suited for 1-bit sparse spike processing, leading to considerable overhead and energy efficiency [8]. Even specialized SNN accelerators, often composed of fewer, larger cores [9–12], encounter substantial challenges when mapping cortical-scale networks [13]. These networks, typically comprising millions to tens of billions of neurons and billions to trillions of synapses,

necessitate distributing neurons and synapses across limited processing units or memory blocks, thereby incurring significant data movement overhead from inter-neuronal communication. This constraint restricts the integration of large-scale parallel processing units, which are indispensable for accurately simulating the inherent parallelism of biologically faithful models or realizing cortical-scale computation [14,15].

A critical factor for emulating the massive intrinsic parallelism of SNNs in near-real-time is achieving a comparable level of hardware parallelism. Biological systems, such as the human cerebral cortex (16–23 billion neurons over $\sim$1800 cm^2), exhibit remarkable computational power within a compact volume, boasting neuron densities around 100 k neurons/mm^2 [16–18]. In stark contrast, current state-of-the-art neuromorphic VLSI chips are severely constrained by low areal density. For instance, IBM's TrueNorth [19,20] integrated 1 million neurons on a large 430 mm^2 die (28 nm), limiting cost-effectiveness. Intel's Loihi series [21–23], while improving integration (Loihi-2: 1 M neurons, 31 mm^2, Intel 4), incorporated complex features like on-chip learning, which added area overhead and constrained pure computational density for inference-focused designs. Other platforms also highlight this density constraint, from custom ASICs like Tianjic [24,25] (39 k neurons, 14.44 mm^2, 28 nm) and MorphIC [26] (2 k neurons, 65 nm), to architectures with different design goals. For instance, the asynchronous mixed-signal DYNAP-SE2 [27] processor prioritizes biological fidelity, implementing 1024 physical neurons on a 98 mm^2 chip (180 nm), while flexible FPGA platforms like SNAVA [28] are designed for research reconfigurability rather than optimized density. Consequently, existing digital neuromorphic chips have largely stagnated around an effective density of approximately 1 k virtual neurons/mm^2 at 40 nm, creating a significant gap compared to their biological counterparts.

Consequently, digital neuromorphic chips have largely stagnated around an effective density of approximately 1 k virtual neurons/mm^2 at 40 nm, a two-orders-of-magnitude gap compared to biological counterparts. This low integration density means that attempting to model cortical-scale networks [13,16–18] or advanced AI models like large-scale spiking Transformers [29,30] on such platforms would necessitate hundreds of chips, leading to impractical systems crippled by inter-chip communication bottlenecks. Conversely, densities exceeding 100 k neurons/mm^2 could enable their integration onto a single die or a few chiplets, making large-scale deployment feasible, especially for size-, weight-, and power-constrained (SWaP) edge applications [31–35]. Boosting on-chip neuron density is, therefore, the foremost challenge for scalable and impactful neuromorphic hardware [36].

Achieving densities approaching the 100 k neurons/mm^2 baseline necessitates radical architectural shifts, primarily the elimination of area-hungry on-chip SRAM for weights, a component known for poor scaling in advanced nodes (e.g., TSMC 7-3nm [37–39]) and a primary density limiter in prior designs. Furthermore, aggressive time-division multiplexing (TDM) is often employed to increase virtual neuron count. However, these strategies introduce significant new VLSI challenges: (1) Static TDM, while increasing virtual neuron capacity, often suffers from poor hardware utilization when mapping diverse network structures, thereby diluting effective density gains, as observed in our previous work [40] where large networks could not be deployed efficiently, and small networks led to idle hardware. (2) Removing SRAM storage shifts a massive data traffic burden to package pins, creating an external-bandwidth wall, inflating I/O energy, and throttling throughput. While any accelerator can support many virtual neurons given sufficient external memory, doing so typically incurs substantial energy and latency penalties for frequent off-chip data access.

This work is an extension of our previous conference work [41], which primarily introduced the structural design and feasibility experiments of HDNIP's flexible network computing. To address these fundamental issues, we introduce HDNIP, a 40 nm high-

density neuromorphic inference platform architecture. HDNIP differs by maximizing the physical density of computational cores on-chip. It integrates 80 flexible-core clusters (FCCs), comprising 1280 parallel cores, capable of hosting up to 10 million neurons and 80 billion synapses within a single die. Our contributions are as follows:

(1) Density-First Architecture: We prioritize computational density by eliminating on-chip weight SRAM, employing a highly reusable core, achieving a synthesis estimated at 225 k neurons/mm^2 (synthesized at 40 nm).

(2) Adaptive Utilization Enhancement: We improved hardware utilization across diverse network topologies compared to static schemes through adaptive TDM and dynamic core fusion, approaching significantly high utilization.

(3) Bandwidth Wall Mitigation: We introduce a ReRAM-based near-memory computing (NMC) strategy for off-chip weight storage, drastically cutting weight bandwidth needs ($\sim$80%). A 4-input-reuse scheme was implemented to achieve a 4-times reduction in input bandwidth, resulting in an overall bandwidth reduction of $\sim$95%. Moreover, the multi-lane interface was optimized to enhance data transmission and facilitate system scalability.

(4) System Validation: We demonstrated capabilities via FPGA emulation and synthesis, achieving 13.64 GSOP/s average throughput for large SNNs at a projected 146 mW, including ReRAM estimates.

In summary, HDNIP addresses the critical limitations of low neuron density, on-chip SRAM bottlenecks, TDM inefficiency, and the external memory bandwidth wall that have constrained previous neuromorphic processors, aiming to provide a promising, scalable path towards monolithic cortex-scale neuromorphic systems.

The remainder of this paper is organized as follows. Section 2 details the proposed HDNIP architecture and its enabling circuit techniques. Section 3 presents the evaluation results. Section 4 concludes and discusses the paper.

2. HDNIP's Architecture

To meet the computational density and throughput requirements for large-scale SNN inference, the overall organizational structure and key modules of the HDNIP architecture have been devised, as illustrated in Figure 1. Without advancing the process node, the architecture achieves large-scale integration of neurons and synapses through a series of density-oriented design strategies. This section delineates HDNIP's architectural details from four perspectives: the overall design philosophy, high-density core array, bandwidth optimization strategy, and a simplified cluster-based communication scheme.

2.1. Overall Organization and Design Philosophy

The overall structure provides the foundation for achieving high computational density, primarily realized through the innovative core design discussed next. The overall architecture of HDNIP is constructed around expanding the core array and simplifying the on-chip support logic. The chip has an extensive parallel core array comprising 1280 configurable computation cores distributed across 80 FCCs, each cluster containing 16 cores. This computation core cluster array enables support for up to 10 million neurons and 80 billion synaptic connections on a single chip, thereby substantially enhancing computational density to a scale approaching biological neural systems. The cores are integrated in a tightly packed arrangement to maximize computational functionality per unit area.

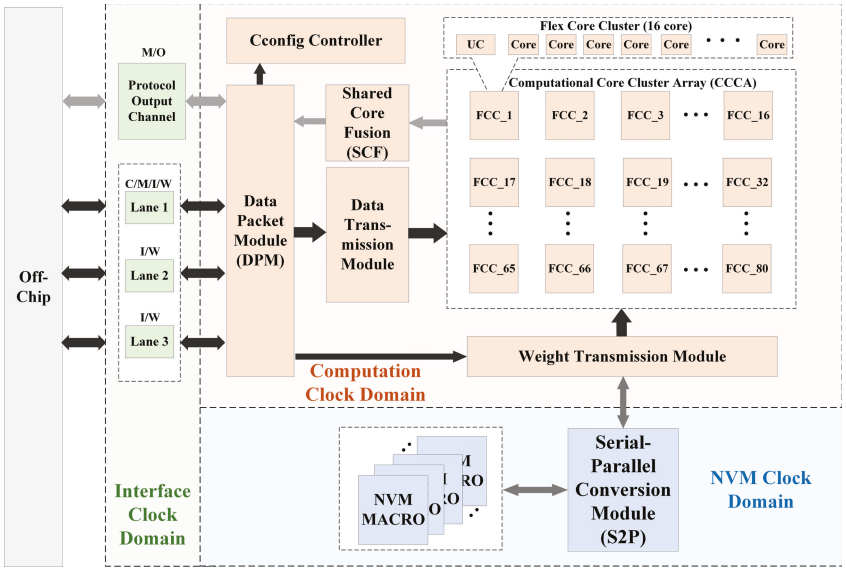


Figure 1. HDNIP's architecture diagram, adapted from [41], reprinted with permission from IEEE Proceedings. ©2024 IEEE.

Moreover, to obviate the need for fine-grained independent control of each core, which would incur additional control overhead and circuit complexity, a cluster-level organizational approach is adopted, whereby every 16 cores are grouped into an FCC managed by a unified controller (UC). This grouping retains the essential core functionalities and communication capabilities while effectively reducing resource overhead related to control and power management, thereby realizing an efficient clustered organizational structure.

Surrounding the core array, HDNIP incorporates a series of lightweight on-chip peripheral modules to provide data transmission and configuration management support. The primary modules include a multi-lane interface for high-speed data input; a configuration controller responsible for global configuration control; a data packet module (DPM) for input data distribution and sequence synchronization; two transmission modules that employ temporary registers for buffering intermediate data and weights; and a serial-to-parallel conversion module (S2P) dedicated to retrieving weight data from external storage. These peripheral modules operate in concert to efficiently interface external storage and transport resources with the internal core array.

Regarding clock design, HDNIP is partitioned into three mutually independent clock domains to balance performance and power consumption across different functional modules: a high-speed computational clock domain drives the core array for neuronal computations; a moderate-frequency, flexibly scalable interface clock domain facilitates chip-to-external communication; and a low-speed non-volatile memory (NVM) clock domain is dedicated to the read and write operations of weight storage devices such as ReRAM. This multi-clock domain design guarantees that each circuit component functions at optimal frequency, minimizing unnecessary power consumption and enhancing overall efficiency.

Importantly, this multi-core architecture, built on configurable cores and organized into FCCs, supports seamless expansion, allowing additional clusters to be integrated with minimal complexity to scale the system further. Although current hardware simulation constraints limit scaling to higher levels, advancements in future process technology and design methodologies are expected to enable chips with even larger neuron scales, ultimately facilitating full cortical simulation.

Thus, through these comprehensive organizational choices and design trade-offs, encompassing the large-scale clustered core array, lightweight peripheral support, and optimized multi-domain clocking, HDNIP establishes the foundational framework necessary

for achieving ultra-high computational density. Building upon this robust foundation, the innovative design and optimization of the core computational units become critical for fully realizing this density and enhancing overall efficiency, as elaborated in the following.

2.2. High-Density-Oriented Versatile Core Design

To maximize the integration of neurons and synapses within the limited silicon area, each HDNIP core minimizes logic and storage overhead. Increasing the TDM factor, although yielding a higher neuron density, reduces deployment flexibility and performance degradation. This section delineates the realization of variable TDM through a compact core implementation and the adoption of cluster sharing to enhance deployment versatility.

2.2.1. Compact Core Design

The 1280 computation cores are divided into 80 core clusters to simplify the scale of hardware control circuits, reduce costs, and lower overhead. As shown in Figure 2a, each flexible core cluster contains 16 cores with consistent virtual neuron array sizes and is controlled by a unified configuration module for the cluster.

Figure 2b shows the circuit design of the compact core, which includes the synapse unit, dendrite unit, and soma unit. In the synapse unit, a weight-sharing method is used. A 4-bit weight state is obtained from outside the core, and the corresponding 16-bit weight data are retrieved from the internal lookup table for computation. TDM saves area overhead in the dendrite unit, where one of the four membrane potentials is accumulated per input data. In the soma unit, leakage, reset, and output generation are performed. The input buffer holds 1-bit values, which, after undergoing accumulation with 16-bit weights, produce a 16-bit accumulated value. Following activation by the soma unit, the output remains a 1-bit value.

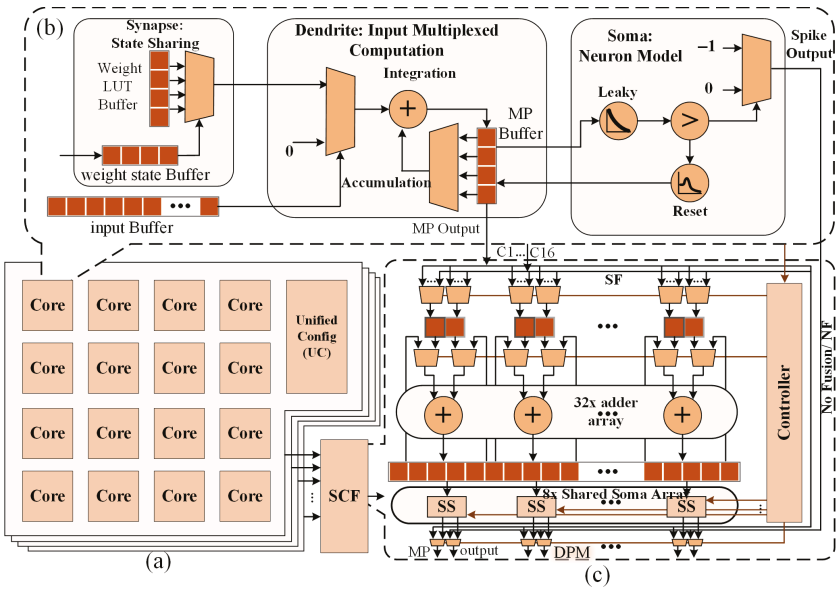


Figure 2. Computation core architecture diagram, including module circuit details. (**a**) The flexible core clusters, adapted from [41], reprinted with permission from IEEE Proceedings. ©2024 IEEE. (**b**) The compact core's circuit design, adapted from [41], reprinted with permission from IEEE Proceedings. ©2024 IEEE. (**c**) The shared core fusion module schematic.

The most popular LIF neuron model is implemented, as shown in the following:

$$\begin{cases} MP_t = MP_{t-1} + \sum W_l(D) S_{t-1} \\ S_t = H(MP_t) \end{cases} \quad (1)$$

where W_l is a weight lookup table (LUT), and when the weight state variable D is inputted, the output weight is calculated. $H(x)$ is a step function: if x is less than 0, no spike is fired; otherwise, a spike is fired. The input buffer in Figure 2b stores spikes. In the dendrite unit, the inverter for weights is disabled, and after the membrane potential calculation, the soma structure in the LIF neuron fires spikes and outputs the membrane potential.

2.2.2. Adaptive Time-Division Multiplexing

Compact cores employ a high-TDM technique in the dendrite unit. Each physical core is time-multiplexed to emulate a virtual neuron–synapse crossbar array that substantially exceeds its inherent hardware scale. In other words, a compact physical core can sequentially compute state updates for neurons over rapid, consecutive temporal segments, effectively expanding the compact core into a large-scale crossbar array. This mechanism facilitates an order-of-magnitude increase in neuron/synapse density, establishing the basis for the architecture's high density.

For medium-to-large-scale networks [29,42], as shown in Figure 3, the number of required synapses typically exceeds 4096, and neurons surpass 8192. By configuring the crossbar scale appropriately, our design better approximates the connectivity observed in the human brain, where each neuron typically connects to 5000–8000 synapses [43]. Our configuration supports up to 8192 neurons per core, allowing the output of one core to serve directly as the synaptic input for the next, which enhances inter-core connectivity.

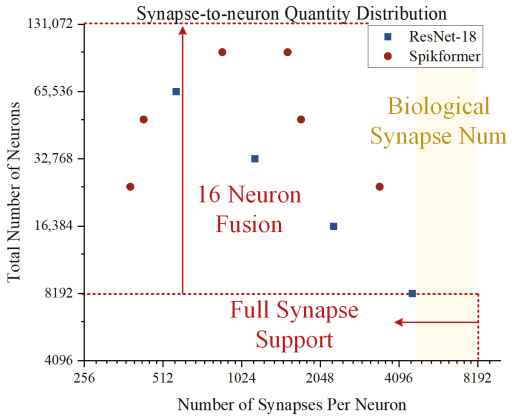


Figure 3. The neuron–synapse scale diagram of ResNet-18 [42] and Transformer [29], where the synapses in the convolution part represent effective synapses, calculated as the product of convolution kernels and input channels.

To improve the flexible deployment capability of high-density computation, the cores within the flexible core cluster support a variable-scale virtual crossbar structure. As shown in Figure 4a, the core can simulate 32–8192 synapses and neurons, and they can be freely combined to form virtual crossbar computation arrays that match the scale of the network mapping. The TDM ratio of each core is not fixed but can be configured before runtime to suit the scale of the targeted neural network layer. When a layer requires a relatively small number of neurons and synapses to be emulated, the number of TDM cycles is reduced to minimize idle overhead; conversely, when the layer's scale approaches the core's maximum capacity, the hardware is fully exploited. Consequently, network layers of varying scales can be executed on a single core with nearly 100% utilization, avoiding resource underutilization inherent in fixed TDM configurations.

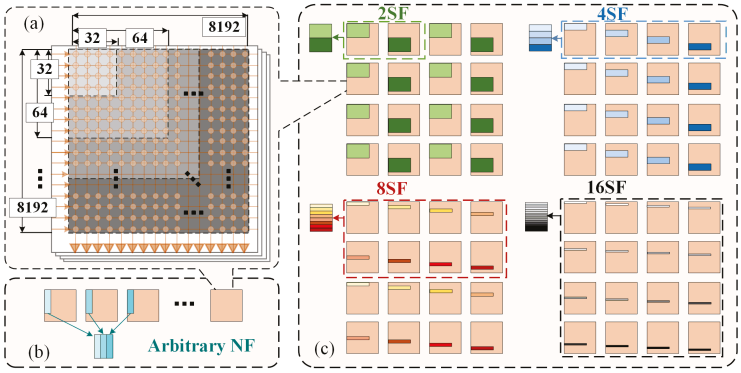


Figure 4. (**a**) Variable-scale core diagram supporting 32–8192, adapted from [41], reprinted with permission from IEEE Proceedings. ©2024 IEEE. (**b**) Core cluster supports arbitrary numbers of neuron fusion (NF). (**c**) Core cluster supports 2, 4, 8, and 16 synapse fusion (SF), represented by green, blue, red, and black, respectively.

2.2.3. Shared Core Fusion

For large network layers that exceed the capacity of a single core, or for small network layers where parallel core computation is desired to reduce latency, HDNIP permits the fusion of multiple cores in the FCC into a single logical computing unit to collaboratively execute the layer's computation. The FCC supports both neuron and synapse splitting and inherently supports neuron-level fusion. The SCF module also supports synapse-level fusion.

Neuron-level fusion (NF), as shown in Figure 4b, is expressed as follows:

$$MP = MP + W_{M\times N}X_N = MP + [\overbrace{W_{m\times N}X_N, \ldots, W_{m\times N}X_N}^{k}] \tag{2}$$

where $m = M/k$. The complete weight matrix is split into k blocks from the output direction, with each block sized $m \times N$, and k supports arbitrary sizes. Since the computation core outputs data only after completing the neuron calculations, the output data are integrated and fused externally without requiring support from the SCF module. As shown on the left of Figure 2c), the soma unit inside the core operates normally, and after bypassing the SCF module, the data are directly output.

For synapse-level fusion, the situation becomes much more complex:

$$MP = MP + W_{M\times N}X_N = MP + \sum^{k} W_{M\times n}X_n \tag{3}$$

where $n = N/k$. The complete weight matrix is split into k blocks from the input direction, with each block sized $M \times n$. Since synapse fusion requires summing the partial results from each core, only a limited number of synapse fusions are supported, as shown in Figure 4c. Synapse fusion cannot support arbitrary scales, and all cores must participate in the fusion process to simplify circuit control. For example, in a 4-neuron core fusion, the results of every 4 cores are fused and output. The summation of partial results and the new output spike generation are integrated into the SCF module. At this point, the soma inside the core ceases operation.

Figure 2c shows the circuit design details of the shared core fusion module. Since the output bandwidth is significantly 32–8192-times lower than the input bandwidth, considering the compact design, the SCF module with pipeline structure is shared among the 80 FCCs. Within the module, to achieve the accumulation and summation of partial results for various types of synapse fusion, an array of 32 adders is constructed. The buffer registers' design allows for the reuse to implement the addition trees of different scales.

In SF mode, the core shown in Figure 2b only outputs partial results, while the result is generated within the SCF. Therefore, an additional shared soma (SS) unit must be incorporated into the module to ensure the correct output of the result.

The decision to share the SCF module across 80 FCCs is a deliberate design choice aimed at maximizing area efficiency without compromising performance at the target scale. This approach is justified for three main reasons. First, the core's output bandwidth is 32–8192-times lower than its input bandwidth, making a single pipelined SCF capable of handling the reduced data flow from all clusters. Second, synapse-level fusion is a specialized operation; having dedicated SCFs for each cluster would result in significant idle hardware and unnecessary area overhead, which contradicts our high-density goal. Finally, while this shared resource could pose a bottleneck in vastly larger systems, our extensive validations confirm that for the 10-million-neuron scale, the SCF does not limit the overall performance, even when deploying large and complex networks.

In summary, the high-density core array of HDNIP, in conjunction with configurable TDM and core fusion techniques, ensures a high neuron integration density per unit area and fully utilizes computational resources through adaptation to varying task scales. Experimental results presented in Section 3 further substantiate deploying a practical large-scale SNN. Compared to conventional fixed TDM, our architecture allows for extremely high utilization, and resource efficiency was further optimized via core fusion, thereby reducing overall latency.

2.3. Bandwidth Optimization Strategy

In pursuit of extremely high density by removing on-chip weight SRAM, HDNIP employs a multifaceted bandwidth optimization strategy to preclude external data transfers from emerging as a system bottleneck.

2.3.1. General Compact-Weight Near-Memory Computing

In large-scale, high-density neuromorphic computing at the 10 M scale, the bandwidth overhead of computation cores is significant. Figure 5a shows the bandwidth proportions of different data types, where the weight overhead far exceeds that of input and MP data. Caching weights can significantly alleviate external bandwidth pressure, enabling near-memory computing and improving computation core utilization.

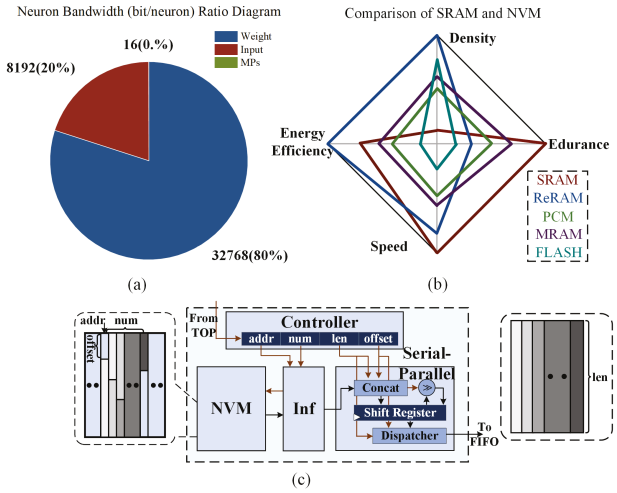


Figure 5. (**a**) Comparison of bandwidth overhead for weights, inputs, and membrane potentials required by each neuron in the designed core. (**b**) Comparison of different types of NVMs and SRAM [44–46]. (**c**) General circuit design for the NVM clock domain, where orange represents the control path, and black represents the data path.

Non-volatile memory has seen rapid development over the past decade. Figure 5b demonstrates that NVMs significantly outperform SRAM in density, which is advantageous for large-scale neuromorphic computing [44–46]. The differences among various NVM types also determine their different applications, but all NVMs generally suffer from lower read–write windows and lower endurance for write–erase cycles [47,48]. This limitation constrains the role of NVMs in tasks requiring frequent read–write operations. Fortunately, the necessary weights for large-scale neuromorphic computing are relatively predictable and do not change with different inputs.

As illustrated in Figure 5b, ReRAM was specifically chosen for this work because it offers the superior density and energy efficiency that are essential for achieving our primary goal of a density-first architecture. While ReRAM is known for having lower write endurance compared to other memories, this was a carefully considered trade-off. This limitation is not a bottleneck for HDNIP because it is an inference-focused platform. In this context, weights are learned offline and written to the memory infrequently. The operational workload is overwhelmingly dominated by read operations, which do not degrade endurance. This "write-once, read-many" usage model effectively neutralizes the impact of low write endurance, a limitation that primarily constrains applications requiring frequent data re-writing.

Therefore, we proposed a general near-memory computing design, primarily intended for frequent NVM read communication, as shown in Figure 5c. Considering the size of the NVM, its bit width, and the fact that the computation core may not be fully active, leading to varying required weight bit widths, the design adopts a flexible configuration to achieve data reading and serial-to-parallel conversion for different scales. The weight data is pre-stored and predictable, allowing it to be prefetched and stored in a FIFO buffer. The serial-to-parallel conversion uses a pipelined merge shift register, supporting arbitrary input and output bit widths while improving throughput. While this approach increases the area overhead, once the input memory bit width is fixed, it can be replaced with a fixed-size design to reduce the overhead.

2.3.2. Input Reuse

Although we utilized NVM to store weight data, significantly reducing the bandwidth demand of the computation core, the statistical results in Figure 5a also indicate that input bandwidth overhead cannot be ignored. Therefore, in our architecture, the data flow of the computation core is modified, as shown in Figure 6a, with a proposed input reuse method to alleviate external bandwidth pressure. For a set of incoming input data, it is not only used for the computation of a single neuron but is repeatedly reused for the computations of different neurons. This means that the core simultaneously calculates the results of multiple neurons.

After adopting the input reuse method, let the reuse factor be r. For the computation of a single core in Figure 6a, the input bandwidth overhead BW_{in} is reduced to

$$BW_{in} = \frac{S \times N}{r} \tag{4}$$

where S represents the number of synapses and N represents the number of neurons. Figure 6b shows the external bandwidth requirements under different values of r, where the input bandwidth is significantly reduced. However, since computations for different neurons are required, the membrane potential data must be cached r times within the core.

Considering the bandwidth pressure illustrated in Figure 6b and the area overhead depicted in Figure 6c resulting from reuse, a 4× reuse factor was chosen to achieve an optimal balance between bandwidth and area overhead.

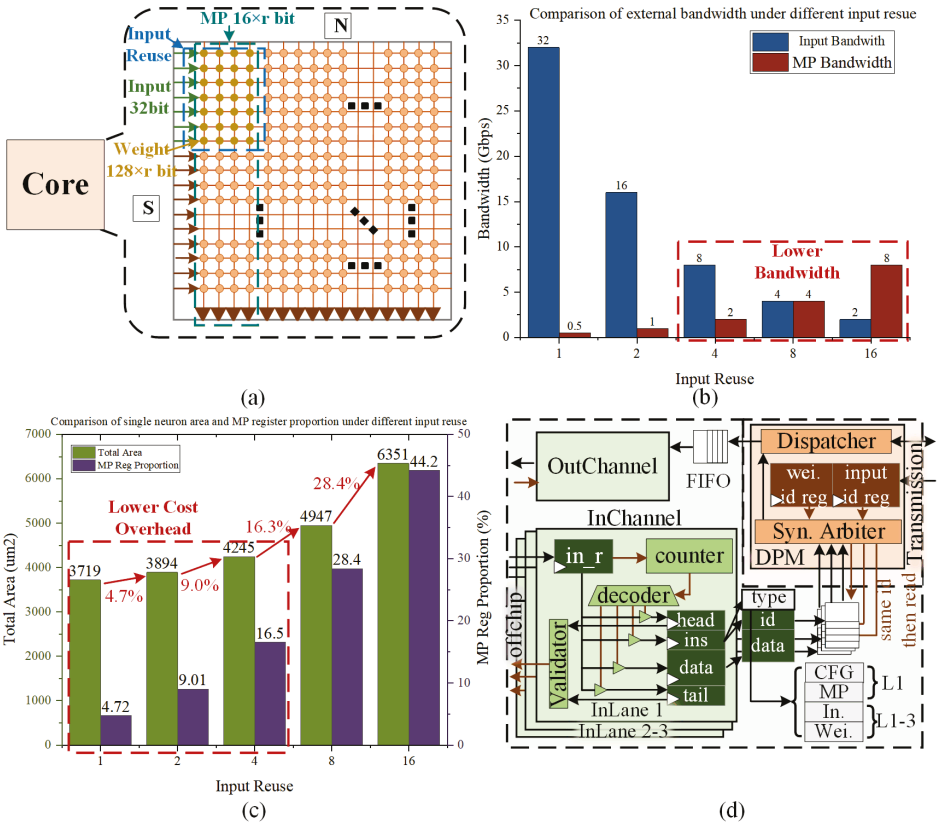


Figure 6. (**a**) Diagram of input reuse in the core. (**b**) Comparison of external bandwidth requirements for the computation core array under different input reuse scenarios at 50 MHz. (**c**) Core gate-level netlist area and membrane potential register area proportion under different reuse scenarios in a 40 nm process. (**d**) Circuit diagram of the multi-lane input protocol in the architecture, where orange arrows represent control paths and black arrows represent data paths.

2.3.3. Multi-Lane Interface

Although the measures have alleviated the bandwidth constraints, high-bandwidth communication between the off-chip memory system and the HDNIP core array remains necessary to satisfy the residual bandwidth requirements and facilitate future expansion, as depicted in Figure 6d. The multi-lane interface comprises three parallel input lanes and a single output channel. Each input lane has a validator and decoder to parse incoming data packets from off-chip, stripping protocol headers/tails, extracting the payload, instruction, and an ID. Based on bandwidth requirements and data type, the lanes are specialized. Configuration and MP data, possessing lower bandwidth needs partially due to the TDM scheme reducing MP update frequency, are exclusively handled by Lane 1. High-bandwidth data, specifically input spikes and synaptic weights, can be transmitted across all three lanes to maximize throughput. Extracted data and associated identifiers are temporarily stored in lane-specific buffers before being processed by the DPM.

A synchronization mechanism is crucial given that input spike and weight data related to the same computational step may arrive asynchronously across the three input lanes. This is implemented via an ID-ACK handshake protocol coordinated between the off-chip data source. The DPM contains internal ID registers holding the expected transaction ID. The Synchronization Arbiter compares the ID of the data waiting in the input lane buffers with the corresponding expected internal ID. Data are only dispatched for computation when the ID matches exactly, effectively stalling lanes with mismatched or subsequent IDs. This ensures that data packets belonging to the same transaction are consumed coherently across the parallel lanes before processing proceeds to the next transaction, maintaining

data integrity. Since CFG and MP data are serialized on Lane 1, they bypass this multi-lane ID synchronization. On the output side, the TDM core design significantly reduces the peak output data rate, allowing a single out channel, buffered by a FIFO, to adequately handle the transmission of both generated spikes and updated membrane potentials off-chip.

Our proposed design leverages NVM for weight storage to reduce external bandwidth overhead by 80%, while the pipelined universal near-memory read design alleviates bandwidth bottlenecks. The 4-input reuse further reduced the primary input spike bandwidth to 25% of its original requirement, resulting in a 95% decrease in overall data volume needing off-chip transmission. The multi-lane interface design, featuring three parallel input lanes and supported by dedicated input buffers and a DPM for distribution, was devised to directly address the increasing bandwidth requirements associated with a growing number of active cores and neurons. The aggregate bandwidth can be linearly expanded by employing these parallel data channels. Furthermore, the ID-ACK mechanism acts as an essential flow control protocol, ensuring data integrity, consistency, and synchronization across these parallel lanes, which is especially critical during concurrent operations across many active cores and prevents data overrun.

These bandwidth optimization strategies operate synergistically, encompassing near-memory computing for weight access, input reuse, a scalable multi-lane interface, and ID-ACK flow control. It ensures that the core array can continuously and reliably access necessary weights and input data to support sustained computation, even as the number of active cores and neurons grows, thereby preventing bottlenecks and supporting scalable performance.

3. Experiments and Results

3.1. Deployment of the Emulation System

We expanded upon our previous experiments [41] and performed detailed functional verification for circuit emulation using the S2C OmniArk emulator, as shown in Figure 7a, composed of 12 Quartus Stratix FPGAs. A deployment of the typical configuration for the HDNIP was implemented on the emulator, as shown in Table 1. We utilized 16 NVM macro arrays, each with a size of 3072 bits $\times$ 2048, totaling 100 Mbits. In the actual deployment on the emulator, DDR is used to store NVM data, serving as an equivalent mapping. The actual synthesized resource overhead is also reflected in Table 1, with 131 k ALMs used and 100.66 Mbits of DDR utilized. The clock frequency ratio of computation, interface, and NVM is 2:8:1. With an NVM bit width of 3072, near-memory computing achieves good throughput without bottlenecks.

In our emulation and verification framework (Figure 7b), the off-chip functionalities are realized within a software-based Runtime Server. This server, implemented primarily in C/C++, interfaces with the Verilog hardware emulator via the DPI-C protocol using SCE-MI. High-level AI model mapping from PyTorch 2.3.1 and driver functions reside in Python 3.10, generating configuration and data tensors deserialized and passed to the C/C++ layer. Within the Runtime Server, the Multi-Lane Hub and Mem Pool module manages the dynamic data flow, effectively simulating the routing and memory interactions discussed previously. Data are formatted using pack/unpack functions and buffered via thread-safe queues before transmission to/from the emulator. Concurrently, the Mem editor component manages synaptic weights generated by the high-level driver to program the emulated NVM.

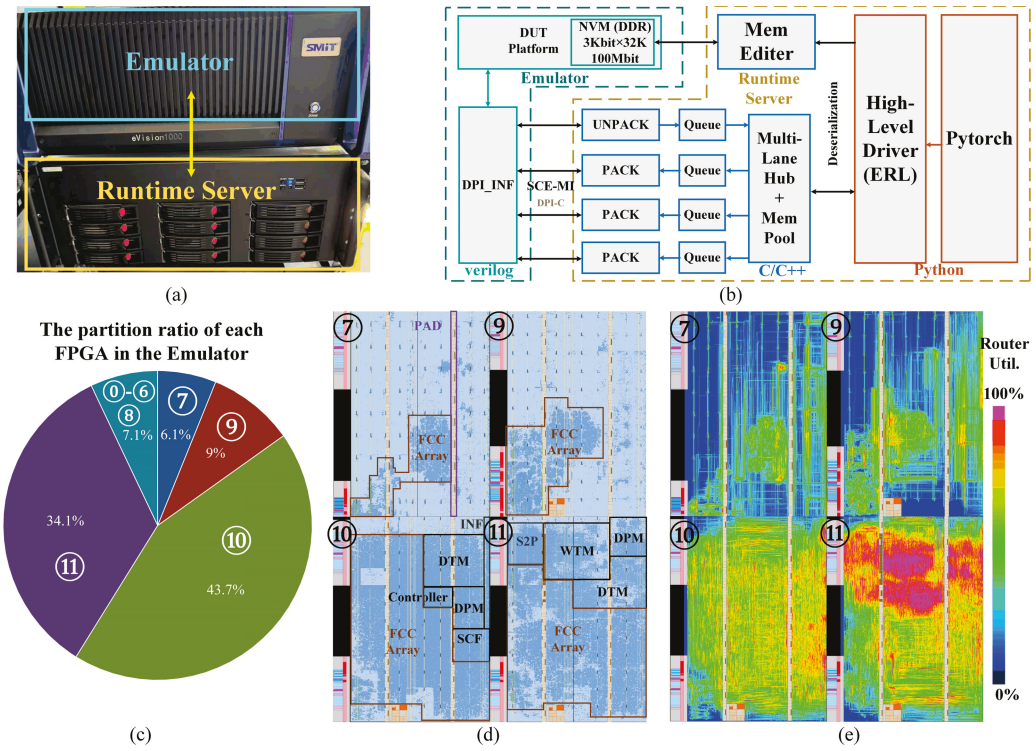


Figure 7. (**a**) Physical diagram of the deployed emulation system, including the emulator and Runtime Server. (**b**) Architecture diagram of the deployed system's software and hardware components, connecting heterogeneous software and hardware data through deserialization and SCE-MI Protocol. (**c**) Partition ratio diagram of FPGA 0–11 on the emulator, where only 7, 9, 10, and 11 are utilized. (**d**) Module placement distribution diagram on the main FPGA board, where dark blue represents the effective resources deployed for the DUT and DTM and WTM are abbreviations for the Data Transmission Module and Weight Transmission Module, respectively. (**e**) Routing congestion level on the main FPGA board, with redder areas indicating higher congestion levels.

Table 1. Configuration parameters and detailed FPGA overhead statistics of the HDNIP on the emulator.

Config. Name	Config. Param.	Emu. Resources	Utilization
Computation Clock	50 MHz	ALM	1,317,449
Interface Clock	200 MHz	LUT	2,634,898
NVM Clock	25 MHz	FF	785,843
NVM Bit Width	3072 bit	BRAM	1,607,168
NVM Depth	2048	LUTRAM	434
NVM Block	16	DDR	100,663,296

The emulation system automatically partitions the HDNIP across 12 FPGAs. Figure 7c–e illustrate how the FPGAs are allocated, placed, and routed within the emulator. For the 10 M-scale architecture, the NVM is mapped using DDR, resulting in only boards 10 and 11 being extensively utilized. In contrast, boards 0–9 are sparsely deployed, with more resources dedicated to routing overhead. The layout of each module is shown in Figure 7d, with 69% of the overhead attributed to the computation core array, represented in brown. Unlike chip layouts and single-FPGA layouts, the layout of multi-FPGA partitions is tightly concentrated near the FPGA pads, as shown in the purple areas in Figure 7d. As a result, the DTM and DPM are partitioned onto boards 10 and 11, while the core cluster arrays

on boards 7 and 9 are also placed near the pads to facilitate communication with other boards. Figure 7e shows the routing congestion levels. This congestion is manageable since the actual bottleneck of the simulation system is caused by PCB communication between different boards and DDR delay.

3.2. Emulation Deployment of Neural Network Architectures

Furthermore, while the platform utilizes a simple neuron model, our validation on complex tasks demonstrates its high capability. As detailed in Table 2, various networks were successfully mapped and deployed on the HDNIP emulation system. This includes the successful deployment of a large-scale Spikingformer on the ImageNet dataset, which proves that network-level architectural complexity can effectively solve challenging problems, even with computationally simple neurons. We argue that massive scale is, in itself, a crucial capability. By providing a single-chip platform with 10 million neurons, HDNIP opens the door to a new class of complex, scale-dependent applications in both AI and computational neuroscience that were previously intractable on a single device.

Table 2. Deployment information of four model architectures for different types of tasks.

Datasets	Architecture	Deployed Layers	Batch Size	FCC Utilized	NMC Throughput
N-MNIST	3FC	3FC	416	78	13.29 GSOP/s
Cifar-10	3Tokenizer +2Transformer	1Transformer	8	80	13.30 GSOP/s
ImageNet	5Tokenizer +8Transformer	8Transformer	1	80	13.59 GSOP/s

3.2.1. N-MNIST

We deployed SNN on the HDNIP to recognize N-MNIST, with a structure consisting of three fully connected layers ($2304 \times 1024 \times 512 \times 10$). The networks were fully mapped onto the platform, with each FCC handling one layer of data for 16 images, using 78 FCCs to recognize 416 data samples. Figure 8a shows the datasets' batch recognition accuracy for 10 k images. LUT and weight states were obtained using layer-wise quantization and K-means clustering. The accuracy of our emulator-simulated network batches is shown by the blue squares in Figure 8a, achieving an average accuracy of 98.45%, which remains excellent compared to the full-precision software recognition accuracy of 98.54%.

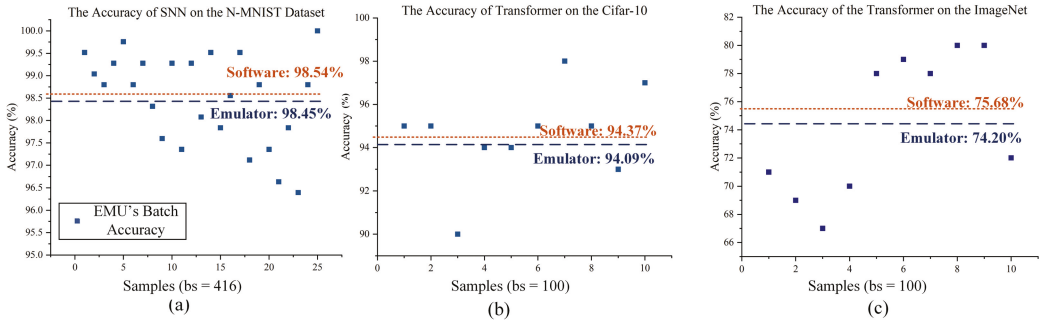


Figure 8. Illustration of inference accuracy for HDNIP across three different datasets with varying random batches. The blue squares represent the final accuracy of individual batch data, while the average accuracy of hardware implementation (blue dashed line) shows only minor differences compared to the full-precision software baseline (orange). (**a**) N-MNIST's accuracy; (**b**) Cifar-10's accuracy; (**c**) ImageNet-1k's accuracy

3.2.2. CIFAR-10

Our work trained a Spikingformer [29] to infer the CIFAR-10 dataset. The Spiking Self-Attention (SSA) module was deployed on the emulator. The number of heads is 12, resulting in the linear layer scale for Query, Key, and Value being 384 × 384. We used a core size of 96 × 32 to deploy the linear layers, utilizing 72 FCCs, with 9 FCCs per layer. The computation for QK^TV was deployed on the remaining 8 FCCs, with a core size of 32 × 8192. The software result is 94.37%, as shown in Figure 9b—orange line. Table 2 shows that only 8 data samples were deployed on the HDNIP, and results for a random selection of 1000 data samples in blue squares indicate that the accuracy did not significantly drop, achieving an accuracy of 94.09%. In contrast to deployed quantized CNN models, Liu et al. [49] achieved an accuracy of 91.53% on a 5.1 M-parameter CNN employing 9-bit quantization on Loihi. At the same time, Tianjic [19,20] attained an accuracy of 93.52% with an int8-quantized VGG-8 model comprising 5.7 M parameters.

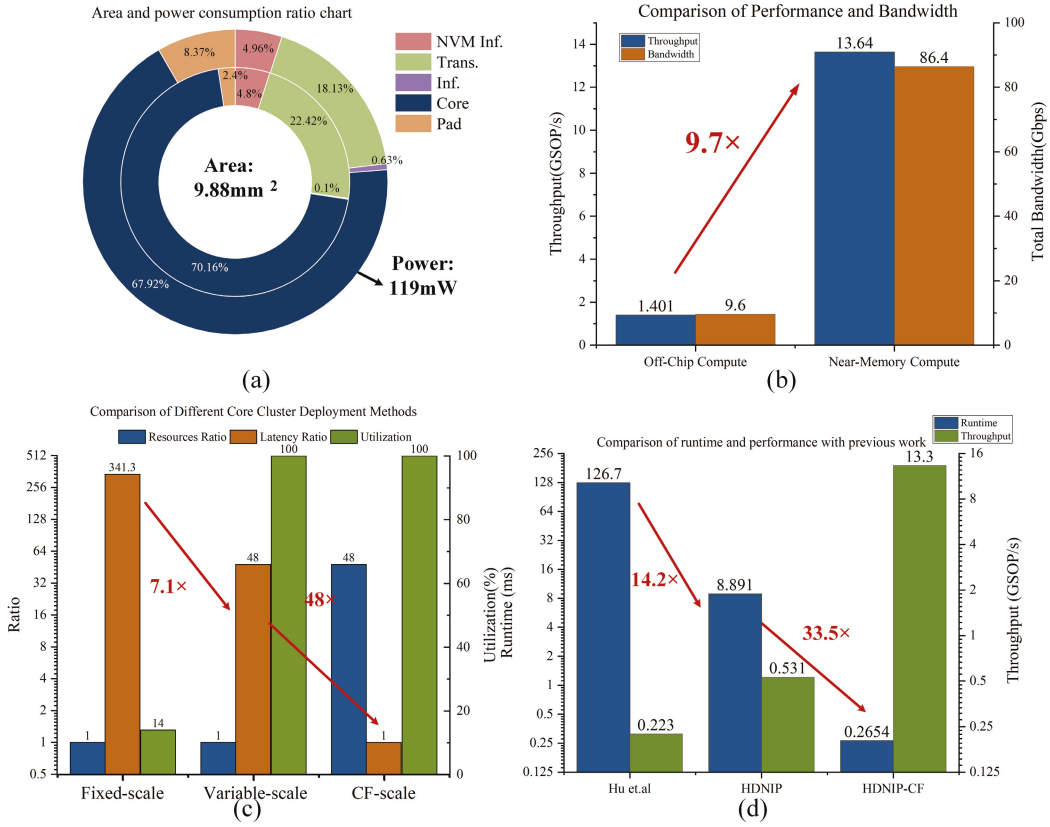


Figure 9. (**a**) A statistical chart of the synthesized area and power consumption proportions, with the inner circle representing the area distribution and the outer circle representing the power consumption distribution. (**b**) Throughput and bandwidth comparison between off-chip storage computation and near-memory computation. (**c**) Overhead comparison of different core cluster deployment methods, adapted from [41], reprinted with permission from IEEE Proceedings. ©2024 IEEE. (**d**) Comparison diagram of the runtime and throughput consumption between Hu et al. (our previous work) and the HDNIP (variable-scale and CF-scale) when executing eight SSA linear layers.

3.2.3. ImageNet

To validate the effectiveness of large-scale data, we employed the dataset from Spikingformer-8-768 to perform deployment validation on the ImageNet-1k dataset. In the emulator deployment, we unfolded the 196 eight-head outputs from the SSA for linear layer computation, with each linear layer consuming 768 × 768 operations. A set of unfolded linear layers utilized 49 cores, each with dimensions of 768 × 3072, and employed a 4-NF

fusion strategy. Additionally, 80 FCCs were deployed to compute the SSA component within eight Transformer layers. The open-source Spikingformer-8-768 model achieved an accuracy of 75.68%, as depicted in Figure 8c. After executing the emulator simulation on a random sample of 1000 images, shown as blue squares in Figure 8c, the HDNIP achieved an accuracy of 74.2%, demonstrating its efficacy.

3.3. Results and Measurements

3.3.1. Chip Synthesies

The architecture proposed in this paper, excluding the NVM, was synthesized using the SMIC 40 nm process at 200 MHz, yielding the results shown in Figure 9a. The total synthesized area of the HDNIP is 9.88 mm^2, with the cores accounting for 70.16%, which is close to the 69% layout proportion seen in the emulator's layout in Figure 7d. Under the same clock frequency as specified in Table 1, near-memory computing operating with 8192 fully multiplexed cores exhibited a power consumption (excluding NVM) of 119.22 mW, with cores accounting for 67.92% of the total. The power consumption of the external interface diluted this proportion.

For the general compact-weight near-memory computing, a 100 Mbits ReRAM memory replaces SRAM. Table 3 shows that the ReRAM IP demonstrates a 1.9-times reduction in area and a 2.25-times reduction in power consumption compared to SRAM, closely aligning with the data reported in other studies [50,51]. Considering the area overhead of the NVM storage, as shown in Table 3, the HDNIP architecture achieves a 1.7-times reduction in area and a 1.2-times reduction in power compared to the SRAM-based version.

Table 3. A comparison of the area and power consumption of 100 Mbits SRAM and ReRAM, both individually and within the context of HDNIP.

Memory Type	Mem. Area (mm^2)	HDNIP Area (mm^2)	Mem. Power (mW)	HDNIP Power (mW)
SRAM	4.13	75.95	61.16	180.38
ReRAM	2.17	44.39	27.15	146.37

3.3.2. NMC and off-chip computing measurements

By incorporating the ReRAM IP, 1280 cores were utilized, thereby realizing an overall throughput of 13.64 GSOP/s with a total power consumption of 146.37 mW. Conversely, in the absence of ReRAM, with weights being supplied from off-chip, overall performance was constrained by a bandwidth bottleneck, yielding a throughput of 1.4 GSOP/s and a power consumption of 83.41 mW. The current evaluation used randomized networks and workloads rather than specific ones. A comprehensive configuration was adopted, with all 1280 cores active, a TDM factor of 8192, and complete consideration of power consumption.

As shown in Figure 9b, the results indicate that the throughput improvement achieved by using NVM near-memory computation is up to 9.7 times, primarily due to a 9.0-times increase in bandwidth. The throughput improvement exceeding 9.0 times is attributed to the reduction in external bandwidth loss, which improved the actual bandwidth utilization. The average maximum energy efficiency ratio is 10.73 pJ/SOP, which is 5.5 times higher than the energy efficiency ratio of off-chip computation. The throughput and energy efficiency improvements of near-memory computing benefit from alleviating external bandwidth bottlenecks. The bandwidth bottleneck is eliminated by adopting general-weight near-memory computing and input reuse designs.

3.3.3. Core Fusion Measurements

HDNIP's flexible architecture, featuring variable-scale cores and core fusion, significantly enhances mapping efficiency compared to fixed-scale designs. For instance, when deploying a 384 × 384 linear layer (e.g., in Spikingformer for Cifar-10), a prior fixed-core approach suffered from very low utilization (14%) and 7.1-times higher latency. In contrast, HDNIP's variable-scale cores achieve nearly 100% utilization. Furthermore, its core fusion mechanism can parallelize computation across 48-times more resources (e.g., using 4-way synapse and 12-way neuron fusion), theoretically reducing latency 48-fold while maintaining 100% utilization.

Cycle-accurate simulations demonstrate these benefits: the previous fixed-scale work [40] took 126.7 ms with a 0.223 GSOP/s throughput for one linear layer. HDNIP with variable-scale cores reduced runtime 14.2-fold and doubled throughput to 0.531 GSOP/s. Employing core fusion across 72 FCCs further slashed runtime to 0.2654 ms (a 477-fold improvement over the previous work) and boosted throughput to 13.3 GSOP/s. While this high parallelism revealed a minor input bottleneck due to the synapse TDM factor of 96 (which a larger TDM could alleviate), the results strongly affirm the substantial performance gains from HDNIP's architectural flexibility.

To quantify the trade-off between density and latency inherent to our high TDM factor, we analyzed the neuron update rate under varying core configurations. The results highlight HDNIP's designed versatility. When mapping smaller, agile networks (e.g., a 96 × 32 virtual array), a core achieves a high update rate of 3380 Hz, suitable for real-time applications, with latency further reducible via our core fusion mechanism. Conversely, when configured for maximum capacity with an 8192 × 8192 array, the rate becomes 0.16 Hz. This lower rate is tailored for high-throughput, non-real-time tasks such as large-scale scientific simulations, where emulating the network's full complexity is prioritized over single-step latency. Thus, HDNIP's adaptive architecture allows it to effectively serve a wide spectrum of applications, from latency-sensitive tasks to massive-scale computational research.

3.4. Comparison

3.4.1. Comparison with Conventional GPU

As mentioned in the introduction, traditional high-performance GPU architectures are inefficient when handling sparse, event-driven spiking neural networks. To quantify this disparity, we compared HDNIP with the top-tier NVIDIA RTX 4090 GPU while executing the Spikingformer (ImageNet) task, as mentioned in Table 4. The results clearly highlight the advantages of a specialized architecture; although the RTX 4090 achieves high performance at 28.12 TSOPS, its power consumption reaches 450 W, resulting in an energy efficiency of only 16.00 pJ/SOPS.

In contrast, HDNIP delivers an exceptional energy efficiency of 10.73 pJ/SOP while consuming just 146.37 mW, outperforming the RTX 4090 by approximately 1.5×. More importantly, HDNIP's power consumption is over three orders of magnitude lower than that of the GPU. This significant advantage in energy efficiency and power consumption validates our core argument: despite GPUs' high computational power, their substantial energy overhead makes them unsuitable for SWaP-constrained edge applications. For deploying large-scale SNNs in energy-limited scenarios, specialized high-density platforms like HDNIP represent a more feasible and efficient solution.

Table 4. Comparison of results between our work and GPU in the Spikingformer.

Platform	This Work (HDNIP)	RTX4090 (GPU)
Technology (nm)	40	4
Power (mW)	146.37	450,000
Throughput (GSOP/s)	13.64	28,121
Energy Efficiency (pJ/SOP)	10.73	16.00

3.4.2. Comparison with other neuromorphic works

The HDNIP achieved 10 million neurons and 80 G synapses using a synthesized area of 9.88 mm^2 without NVM's area and 44.39 mm^2 with 100 Mbits ReRAM. We use the number of virtual neurons $N_{neu.}$ that can be deployed per unit area as a measure of the architecture's density, as shown below:

$$\rho_{neu.} = \frac{N_{neu.}}{Area} \tag{5}$$

Table 5 indicates whether HDNIP incorporates NVM, and when normalized to the same process node, its neuron density outperforms other works by 2–3 orders of magnitude. As illustrated in Figure 10, the neuron density of other works normalized to a 40 nm process is approximately 1 K/mm^2. In contrast, this design's HDNIP achieves a 225–1012 K/mm^2 density, which is remarkably 71–320-times greater than the results of our previous work [40] and far ahead of other works. This approach involved not only architectural choices, such as removing low-density SRAM and employing a highly reusable core, but also a deliberate focus on inference by implementing a computationally efficient neuron model and omitting area-intensive features like on-chip learning. Additionally, the synthesized area, rather than the chip layout and routing area, makes the density improvement very significant. However, the density improvement remains highly substantial even when accounting for layout, routing, and more NVM overhead. HDNIP's neuron density is already two-times higher than our proposed high-density baseline, offering promise for deploying brain-scale, human-brain-like simulations on a single device with SOTA technology.

Table 5 reports throughput measured at the maximum sustainable clock (or equivalent asynchronous scan rate) with all virtual neurons active. The proposed HDNIP integrates 1280 cores and operates at a 50 MHz, achieving 13.64 GSOP/s. In contrast, the design reported by Hu et al. [40] incorporates nine cores operating at 100 MHz, yielding 0.22 GSOP/s, while MorphIC [26] implements four cores operating at 55 MHz to achieve 0.11 GSOP/s. Moreover, TrueNorth [52] features 4096 asynchronous cores and, when normalized to an equivalent 65 MHz scan frequency, sustains 2.99 GSOP/s.

Table 5 reports two harmonized metrics: energy per synaptic operation (pJ/SOP) and normalized compute density (MSOPS/mm^2) to facilitate a fair cross-chip assessment. In contrast to neuron density, which represents the network deployment capacity per unit area, computational density denotes the throughput per unit area. Our HDNIP achieves leading performance with 10.7 pJ/SOP and 307 MSOPS/mm^2, significantly outperforming our previous work [40] (33.26 pJ/SOP, 69 MSOPS/mm^2) by offering an efficiency ratio of 3 times and computational density of 4.4 times. In terms of energy efficiency, the HDNIP architecture exhibits certain deficiencies relative to Tianjic's work [24,25], yet outperforms other approaches; moreover, it achieves a 3.7× improvement in computational density over MorphIC [26]. Therefore, the HDNIP architecture achieves high neural density and, through architectural optimization, attains elevated performance; compared with other works, it exhibits a competitive energy efficiency ratio and superior computational density.

Table 5. Results of the architecture in this paper and comparison with similar works.

Platform	This Work	Hu et al. [40]	True-North [19,20]	Loihi-1 [21]	Loihi-2 [22,23]	Tianjic [24,25]	Morph-IC [26]	DYN-AP-SE2 [27]	SNA-VA [28]
Technology (nm)	40	55	28	14	7	28	65	180	28
Clock (MHz)	25–200 [a]	100	Async	Async	Async	300	55/210	Async	125
Neuron Number	10 M	10 k	1 M	128 k	1 M	39 k	2 k	1 k	12.8 k
Synapse Number	80 G	10 M	256 M	128 M	114.4 M	9.75 M	2.06 M	65 k	20 k
Area (mm^2)	44.39	6	430	60	31	14.44	3.5	98	12
Normalized neuron density (K/mm^2) [b]	225.28	3.15	1.14	0.26	0.99	1.32	1.5	0.21	0.52
Virtual Neurons Per Core	8192	1024	256	1024	8192	256	512	1	1024
Core Number	1280	9	4096	128	128	156	4	4	100
Power (mW)	146.37	14.9	42–323	-	-	950	19.9	-	625
Avg. Through-put (GSOP/s)	13.64	0.22	2.99	-	-	-	0.11	-	-
Energy Efficiency (pJ/SOP)	10.73	33.26	21.73	23.6	-	1.54	51	150	-
Normalized Compute Density (MSOPS $/mm^2$) [c]	307	69	34	-	-	-	83	-	-

[a] The interface clock operates at 200 MHz, the computation clock at 50 MHz, and the NVM clock at 25 MHz, the same as Table 1. [b] The area is normalized to 40 nm, $Area_{\text{Normalized}} = Area \times \left(\frac{\text{Tech.}}{40}\right)^2$. [c] The computational density: $\frac{Throughpput}{Area_{Normalized}}$.

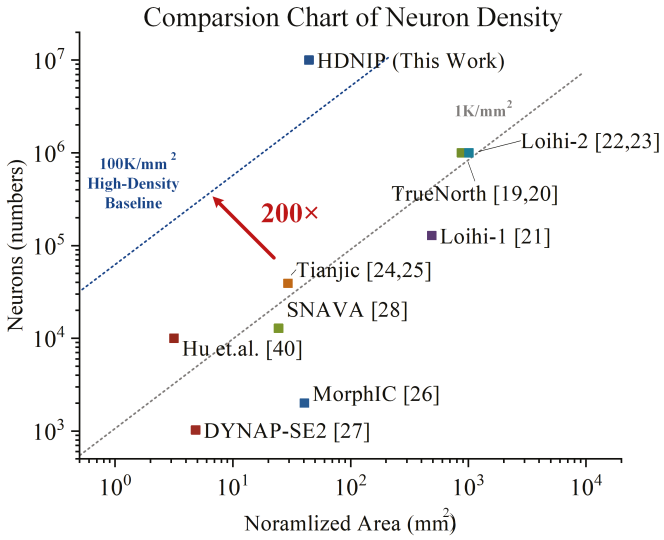


Figure 10. A comparison chart of normalized area (under 40 nm process) versus neuron number across different works.

4. Conclusions and Discussion

In conclusion, this work presents a high-density neuromorphic inference platform that surmounts key hurdles in scaling neuromorphic hardware. The proposed architecture combines several novel strategies: elimination of on-chip SRAM, ReRAM-based NMC, input reuse, and adaptive TDM with core fusion, boosting neuron integration density, performance, and utilization efficiency beyond what prior designs have achieved. Through these techniques, HDNIP delivers a single-chip capacity of up to 10 million spiking neurons and 80 billion synapses, reaching 225 K/mm^2 and demonstrating effective inferencing throughput on the 13 GSOP/s at 146 mW power. Regarding neuronal density, neuromorphic processors have improved by two orders of magnitude compared to the most advanced neuromorphic processors, effectively bridging the long-standing density gap between silicon neural engines and biological engines.

The HDNIP architecture demonstrates that a density-first design can achieve high performance by leveraging advanced memory technologies like ReRAM and carefully considered architecture-level adaptations. Its design is particularly well suited for large-scale spiking neural network inference in energy-constrained scenarios, where maximizing neuron numbers and energy efficiency is paramount. A key aspect of this density-first approach was the intentional focus on inference, utilizing a computationally efficient LIF neuron model while omitting area-intensive on-chip learning mechanisms like STDP. Crucially, this simplification at the neuron level does not limit the platform to simple problems; our successful deployment of complex models like the Spikingformer on the ImageNet dataset demonstrates that network-level architectural complexity can effectively address challenging, large-scale tasks. The platform's adaptive TDM is engineered to effectively manage resource utilization and throughput for its target SNN workloads. While the inherent nature of time-multiplexing might require careful consideration for applications with exceptionally stringent, hard real-time constraints, this is a common trade-off for the significant density achieved. Furthermore, reliance on an off-chip interface is addressed through a scalable multi-lane design that allows for future external high-throughput interface (e.g., NVLink [53], SerDes [54]), expected to provide enough bandwidth to support even cortex-scale network models.

The design experience with HDNIP offers valuable insights for the field: the considerable potential of aggressive on-chip memory reduction, the importance of co-designing computation with advanced memory solutions, and the benefits of architectural flexibility. Building upon the high-density foundation established by HDNIP, our future work will focus on integrating on-chip learning capabilities, such as STDP, and more complex neuron dynamics, exploring novel circuit and architectural techniques to balance the trade-off between neuron density and functional complexity. Overall, HDNIP represents a notable step in neuromorphic VLSI, suggesting a viable direction for embedding many neurons on a single chip and inspiring future architectures to continue advancing towards bio-inspired neural computation.

Author Contributions: Conceptualization, Y.Z. and N.N.; methodology, Y.Z.; software, L.M., R.M.; validation, K.C., R.Z. and C.F.; formal analysis, S.W.; investigation, G.Q.; resources, Y.L.; writing—original draft preparation, Y.Z. and S.H.; writing—review and editing, Y.Z. and S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by STI 2030-Major Projects 2022ZD0209700 and in part by the Sichuan Provincial Science and Technology Programs (Grant No. 2024ZDZX0001 and Grant No. 2024ZYD0253).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors gratefully acknowledge the support provided by the University of Electronic Science and Technology of China (UESTC).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Mehonic, A.; Kenyon, A.J. Brain-inspired computing needs a master plan. *Nature* **2022**, *604*, 255–260. [CrossRef]
2. Li, G.; Deng, L.; Tang, H.; Pan, G.; Tian, Y.; Roy, K.; Maass, W. Brain-Inspired Computing: A Systematic Survey and Future Trends. *Proc. IEEE* **2024**, *112*, 544–584. [CrossRef]
3. Zhang, W.; Ma, S.; Ji, X.; Liu, X.; Cong, Y.; Shi, L. The development of general-purpose brain-inspired computing. *Nat. Electron.* **2024**, *7*, 954–965. [CrossRef]
4. Shrestha, A.; Fang, H.; Mei, Z.; Rider, D.P.; Wu, Q.; Qiu, Q. A survey on neuromorphic computing: Models and hardware. *IEEE Circuits Syst. Mag.* **2022**, *22*, 6–35. [CrossRef]
5. Furber, S. Large-scale neuromorphic computing systems. *J. Neural Eng.* **2016**, *13*, 051001. [CrossRef]
6. Jouppi, N.P.; Young, C.; Patil, N.; Patterson, D.; Agrawal, G.; Bajwa, R.; Bates, S.; Bhatia, S.; Boden, N.; Borchers, A. In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), Toronto, ON, Canada, 24–28 June 2017; pp. 1–12.
7. Tirumala, A.; Wong, R. Nvidia blackwell platform: Advancing generative ai and accelerated computing. In Proceedings of the 2024 IEEE Hot Chips 36 Symposium (HCS), Stanford, CA, USA, 25–27 August 2024; pp. 1–33.
8. Rathi, N.; Chakraborty, I.; Kosta, A.; Sengupta, A.; Ankit, A.; Panda, P.; Roy, K. Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware. *ACM Comput. Surv.* **2023**, *55*, 1–49. [CrossRef]
9. Basu, A.; Deng, L.; Frenkel, C.; Zhang, X. Spiking Neural Network Integrated Circuits: A Review of Trends and Future Directions. In Proceedings of the 2022 IEEE Custom Integrated Circuits Conference (CICC), Newport Beach, CA, USA, 24–27 April 2022; pp. 1–8. [CrossRef]
10. Kuang, Y.; Cui, X.; Wang, Z.; Zou, C.; Zhong, Y.; Liu, K.; Dai, Z.; Yu, D.; Wang, Y.; Huang, R. ESSA: Design of a Programmable Efficient Sparse Spiking Neural Network Accelerator. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2022**, *30*, 1631–1641. [CrossRef]
11. Tan, P.Y.; Wu, C.W. A 40-nm 1.89-pJ/SOP Scalable Convolutional Spiking Neural Network Learning Core With On-Chip Spatiotemporal Back-Propagation. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2023**, *31*, 1994–2007. [CrossRef]
12. Xie, C.; Shao, Z.; Chen, Z.; Du, Y.; Du, L. An Energy-Efficient Spiking Neural Network Accelerator Based on Spatio-Temporal Redundancy Reduction. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2024**, *32*, 782–786. [CrossRef]
13. Cadwell, C.R.; Bhaduri, A.; Mostajo-Radji, M.A.; Keefe, M.G.; Nowakowski, T.J. Development and arealization of the cerebral cortex. *Neuron* **2019**, *103*, 980–1004. [CrossRef]
14. Ballard, D.H. Cortical connections and parallel processing: Structure and function. *Behav. Brain Sci.* **1986**, *9*, 67–90. [CrossRef]
15. Mosier, K.; Bereznaya, I. Parallel cortical networks for volitional control of swallowing in humans. *Exp. Brain Res.* **2001**, *140*, 280–289. [CrossRef] [PubMed]
16. Herculano-Houzel, S. The human brain in numbers: A linearly scaled-up primate brain. *Front. Hum. Neurosci.* **2009**, *3*, 31. [CrossRef] [PubMed]
17. Azevedo, F.A.; Carvalho, L.R.; Grinberg, L.T.; Farfel, J.M.; Ferretti, R.E.; Leite, R.E.; Filho, W.J.; Lent, R.; Herculano-Houzel, S. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *J. Comp. Neurol.* **2009**, *513*, 532–541. [CrossRef]
18. Pakkenberg, B.; Gundersen, H.J.G. Neocortical neuron number in humans: Effect of sex and age. *J. Comp. Neurol.* **1997**, *384*, 312–320. [CrossRef]
19. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673. [CrossRef]
20. Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Merolla, P.; Imam, N.; Nakamura, Y.; Datta, P.; Nam, G.J. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2015**, *34*, 1537–1557. [CrossRef]
21. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99. [CrossRef]

22. Orchard, G.; Frady, E.P.; Rubin, D.B.D.; Sanborn, S.; Shrestha, S.B.; Sommer, F.T.; Davies, M. Efficient neuromorphic signal processing with loihi 2. In Proceedings of the 2021 IEEE Workshop on Signal Processing Systems (SiPS), Coimbra, Portugal, 20–22 October 2021; pp. 254–259.
23. Davies, M. Taking neuromorphic computing to the next level with Loihi2. *Intel Labs' Loihi* **2021**, *2*, 1–7.
24. Pei, J.; Deng, L.; Song, S.; Zhao, M.; Zhang, Y.; Wu, S.; Wang, G.; Zou, Z.; Wu, Z.; He, W. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **2019**, *572*, 106–111. [CrossRef]
25. Deng, L.; Wang, G.; Li, G.; Li, S.; Liang, L.; Zhu, M.; Wu, Y.; Yang, Z.; Zou, Z.; Pei, J. Tianjic: A unified and scalable chip bridging spike-based and continuous neural computation. *IEEE J. -Solid-State Circuits* **2020**, *55*, 2228–2246. [CrossRef]
26. Frenkel, C.; Legat, J.D.; Bol, D. MorphIC: A 65-nm 738k-Synapse/mm^2 Quad-Core Binary-Weight Digital Neuromorphic Processor With Stochastic Spike-Driven Online Learning. *IEEE Trans. Biomed. Circuits Syst.* **2019**, *13*, 999–1010. [CrossRef]
27. Richter, O.; Wu, C.; Whatley, A.M.; Köstinger, G.; Nielsen, C.; Qiao, N.; Indiveri, G. DYNAP-SE2: A scalable multi-core dynamic neuromorphic asynchronous spiking neural network processor. *Neuromorphic Comput. Eng.* **2024**, *4*, 014003. [CrossRef]
28. Sripad, A.; Sanchez, G.; Zapata, M.; Pirrone, V.; Dorta, T.; Cambria, S.; Marti, A.; Krishnamourthy, K.; Madrenas, J. SNAVA—A real-time multi-FPGA multi-model spiking neural network simulation architecture. *Neural Netw.* **2018**, *97*, 28–45. [CrossRef] [PubMed]
29. Zhou, C.; Yu, L.; Zhou, Z.; Ma, Z.; Zhang, H.; Zhou, H.; Tian, Y. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv* **2023**, arXiv:2304.11954.
30. Zhu, R.J.; Zhao, Q.; Li, G.; Eshraghian, J.K. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv* **2023**, arXiv:2302.13939.
31. Harbour, S.; Sears, B.; Schlager, S.; Kinnison, M.; Sublette, J.; Henderson, A. Real-time vision-based control of swap-constrained flight system with intel loihi 2. In Proceedings of the 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), Barcelona, Spain, 1–5 October 2023; pp. 1–6.
32. Perryman, N.; Wilson, C.; George, A. Evaluation of xilinx versal architecture for next-gen edge computing in space. In Proceedings of the 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2023; pp. 1–11.
33. Shamwell, E.J.; Nothwang, W.D.; Perlis, D. A deep neural network approach to fusing vision and heteroscedastic motion estimates for low-SWaP robotic applications. In Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Republic of Korea, 16–18 November 2017; pp. 56–63.
34. Neuman, S.M.; Plancher, B.; Duisterhof, B.P.; Krishnan, S.; Banbury, C.; Mazumder, M.; Prakash, S.; Jabbour, J.; Faust, A.; de Croon, G.C.; et al. Tiny robot learning: Challenges and directions for machine learning in resource-constrained robots. In Proceedings of the 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Incheon, Republic of Korea, 13–15 June 2022; pp. 296–299.
35. Kudithipudi, D.; Schuman, C.; Vineyard, C.M.; Pandit, T.; Merkel, C.; Kubendran, R.; Aimone, J.B.; Orchard, G.; Mayr, C.; Benosman, R. Neuromorphic computing at scale. *Nature* **2025**, *637*, 801–812. [CrossRef]
36. Chicca, E.; Stefanini, F.; Bartolozzi, C.; Indiveri, G. Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems. *Proc. IEEE* **2014**, *102*, 1367–1388. [CrossRef]
37. Wu, S.Y.; Lin, C.; Chiang, M.; Liaw, J.; Cheng, J.; Yang, S.; Tsai, C.; Chen, P.; Miyashita, T.; Chang, C.; et al. A 7nm CMOS platform technology featuring 4 th generation FinFET transistors with a 0.027 μm 2 high density 6-T SRAM cell for mobile SoC applications. In Proceedings of the 2016 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 3–7 December 2016; pp. 2–6.
38. Yeap, G.; Lin, S.; Chen, Y.; Shang, H.; Wang, P.; Lin, H.; Peng, Y.; Sheu, J.; Wang, M.; Chen, X.; et al. 5nm cmos production technology platform featuring full-fledged euv, and high mobility channel finfets with densest 0.021 μm 2 sram cells for mobile soc and high performance computing applications. In Proceedings of the 2019 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 7–11 December 2019; pp. 36–37.
39. Wu, S.Y.; Chang, C.H.; Chiang, M.; Lin, C.; Liaw, J.; Cheng, J.; Yeh, J.; Chen, H.; Chang, S.; Lai, K.; et al. A 3nm CMOS FinFlex™ platform technology with enhanced power efficiency and performance for mobile SoC and high performance computing applications. In Proceedings of the 2022 International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 3–7 December 2022; pp. 27.5.1–27.5.4.
40. Hu, S.; Qiao, G.; Liu, X.; Liu, Y.; Zhang, C.; Zuo, Y.; Zhou, P.; Liu, Y.; Ning, N.; Yu, Q. A Co-Designed Neuromorphic Chip With Compact (17.9 KF 2) and Weak Neuron Number-Dependent Neuron/Synapse Modules. *IEEE Trans. Biomed. Circuits Syst.* **2022**, *16*, 1250–1260. [CrossRef]
41. Zuo, Y.; Ning, N.; Cao, K.; Zhang, R.; Wang, S.; Meng, L.; Qiao, G.; Liu, Y.; Hu, S. Design of a Highly Flexible Hybrid Neural Network Inference Platform with 10 Million Neurons. In Proceedings of the 2024 IEEE Biomedical Circuits and Systems Conference (BioCAS), Xi'an, China, 24–26 October 2024; pp. 1–5.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

43. Tang, Y.; Nyengaard, J.R.; De Groot, D.M.G.; Gundersen, H.J.G. Total regional and global number of synapses in the human brain neocortex. *Synapse* **2001**, *41*, 258–273. [CrossRef]
44. Wang, Z.; Song, Y.; Zhang, G.; Luo, Q.; Xu, K.; Gao, D.; Yu, B.; Loke, D.; Zhong, S.; Zhang, Y. Advances of embedded resistive random access memory in industrial manufacturing and its potential applications. *Int. J. Extrem. Manuf.* **2024**, *6*, 032006. [CrossRef]
45. Chen, A. A review of emerging non-volatile memory (NVM) technologies and applications. *Solid-State Electron.* **2016**, *125*, 25–38. [CrossRef]
46. Kargar, S.; Nawab, F. Challenges and future directions for energy, latency, and lifetime improvements in NVMs. *Distrib. Parallel Databases* **2023**, *41*, 163–189. [CrossRef]
47. Strenz, R. Review and outlook on embedded NVM technologies–from evolution to revolution. In Proceedings of the 2020 IEEE International Memory Workshop (IMW), Dresden, Germany, 17–20 May 2020; pp. 1–4.
48. Kim, S.; Yoo, H.J. An Overview of Computing-in-Memory Circuits with DRAM and NVM. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *71*, 1626–1631. [CrossRef]
49. Liu, S.; Yi, Y. Unleashing Energy-Efficiency: Neural Architecture Search without Training for Spiking Neural Networks on Loihi Chip. In Proceedings of the 2024 25th International Symposium on Quality Electronic Design (ISQED), San Francisco, CA, USA, 3–5 April 2024; pp. 1–7.
50. Zhao, L.; Chen, Z.; Manea, D.; Li, S.; Li, J.; Zhu, Y.; Sui, Z.; Lu, Z. Highly reliable 40nm embedded dual-interface-switching RRAM technology for display driver IC applications. In Proceedings of the 2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), Honolulu, HI, USA, 12–17 June 2022; pp. 316–317.
51. Prabhu, K.; Gural, A.; Khan, Z.F.; Radway, R.M.; Giordano, M.; Koul, K.; Doshi, R.; Kustin, J.W.; Liu, T.; Lopes, G.B. CHIMERA: A 0.92-TOPS, 2.2-TOPS/W edge AI accelerator with 2-MByte on-chip foundry resistive RAM for efficient training and inference. *IEEE J. Solid-State Circuits* **2022**, *57*, 1013–1026. [CrossRef]
52. Cassidy, A.S.; Alvarez-Icaza, R.; Akopyan, F.; Sawada, J.; Arthur, J.V.; Merolla, P.A.; Datta, P.; Tallada, M.G.; Taba, B.; Andreopoulos, A.; et al. Real-time scalable cortical computing at 46 giga-synaptic OPS/watt with ~100× speedup in time-to-solution and ~100,000× reduction in energy-to-solution. In Proceedings of the SC'14: International Conference for High Performance Computing, Networking, Storage and Analysis, New Orleans, LA, USA, 16–21 November 2014; pp. 27–38.
53. Wei, Y.; Huang, Y.C.; Tang, H.; Sankaran, N.; Chadha, I.; Dai, D.; Oluwole, O.; Balan, V.; Lee, E. 9.3 NVLink-C2C: A coherent off package chip-to-chip interconnect with 40Gbps/pin single-ended signaling. In Proceedings of the 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 19–23 February 2023; pp. 160–162.
54. Segal, Y.; Laufer, A.; Khairi, A.; Krupnik, Y.; Cusmai, M.; Levin, I.; Gordon, A.; Sabag, Y.; Rahinski, V.; Ori, G.; et al. A 1.41 pJ/b 224Gb/s PAM-4 SerDes receiver with 31dB loss compensation. In Proceedings of the 2022 IEEE International Solid-State Circuits Conference (ISSCC), Francisco, CA, USA, 20–26 February 2022; Volume 65, pp. 114–116.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Mitigating Context Bias in Vision–Language Models via Multimodal Emotion Recognition

Constantin-Bogdan Popescu, Laura Florea and Corneliu Florea *

Image Processing and Analysis Laboratory (LAPI), National University of Science and Technology Politehnica Bucharest, Splaiul Independenței 313, 060042 Bucharest, Romania; bogdan.popescu1802@upb.ro (C.-B.P.); laura.florea@upb.ro (L.F.)

* Correspondence: corneliu.florea@upb.ro

Abstract: Vision–Language Models (VLMs) have become key contributors to the state of the art in contextual emotion recognition, demonstrating a superior ability to understand the relationship between context, facial expressions, and interactions in images compared to traditional approaches. However, their reliance on contextual cues can introduce unintended biases, especially when the background does not align with the individual's true emotional state. This raises concerns for the reliability of such models in real-world applications, where robustness and fairness are critical. In this work, we explore the limitations of current VLMs in emotionally ambiguous scenarios and propose a method to overcome contextual bias. Existing VLM-based captioning solutions tend to overweight background and contextual information when determining emotion, often at the expense of the individual's actual expression. To study this phenomenon, we created synthetic datasets by automatically extracting people from the original images using YOLOv8 and placing them on randomly selected backgrounds from the Landscape Pictures dataset. This allowed us to reduce the correlation between emotional expression and background context while preserving body pose. Through discriminative analysis of VLM behavior on images with both correct and mismatched backgrounds, we find that in 93% of the cases, the predicted emotions vary based on the background—even when models are explicitly instructed to focus on the person. To address this, we propose a multimodal approach (named BECKI) that incorporates body pose, full image context, and a novel description stream focused exclusively on identifying the emotional discrepancy between the individual and the background. Our primary contribution is not just in identifying the weaknesses of existing VLMs, but in proposing a more robust and context-resilient solution. Our method achieves up to 96% accuracy, highlighting its effectiveness in mitigating contextual bias.

Keywords: emotion recognition; contextual bias; vision language models (VLMs); body pose estimation; multimodal learning; affective computing

1. Introduction

Today, due to the rapid adoption of intelligent systems and the fast growth of powerful artificial intelligence applications, tasks like emotion recognition have become essential in fields such as marketing where understanding consumer sentiment drives strategy affective robotics, enabling more natural human–robot interaction, personalized education, adapting learning experiences to emotional states for more engaging content, digital entertainment, and mental health, aiding in empathetic support systems.

These varied applications demand emotion recognition systems that are both accurate and context-aware. As a result, researchers have turned to more sophisticated machine

learning models that can understand not only facial features but also social and environmental cues that influence human affect. This has led to the growing use of Vision–Language Models (VLMs) in this domain.

Modern approaches increasingly rely on Vision–Language Models (VLMs) [1]. Being efficient zero-shot learners for image captioning [2,3], they provide a deeper understanding of visual scenes by integrating facial expressions, environmental cues, and social context. This enables a significant leap over classical models, which often focused exclusively on facial features and overlooked the broader context in which emotions are expressed.

In emotion recognition from images, context often serves as a powerful cue [4]. However, the sensitivity of VLMs to contextual information introduces vulnerabilities, particularly when the context does not align with the individual's actual emotion. As we will show, in synthetic images (designed to reveal this bias), ambiguous scenes, or clinical settings where expressed emotions may contradict the visual context, VLMs' over-reliance on context becomes a weakness. These inconsistencies raise concerns about the generalization ability of current VLM-based emotion recognition systems, as this dependence can lead to biased or inaccurate predictions posing challenges for reliable human-centric applications.

Multiple works [5,6] have shown that Large Language Models (LLMs) such as GPT, Gemini, or Llama are powerful tools for emotion recognition and are increasingly used to interpret emotions in textual discussions due to their ability to generate empathetic responses that approach human-like quality. Despite their impressive capabilities, these models present some limitations, including tendencies toward overgeneralization biases and a cautious avoidance of negative or culturally sensitive formulations in their responses [7,8]. These findings suggest that large models should be used with caution when employed as key components in emotion recognition systems.

However, recent research suggests that Vision–Language Models may also possess the capacity to verify and refine their own outputs via internal reflection mechanisms. Vision Large Models (VLMs) and Large Language Models (LLMs) share a common architectural foundation (such as a Transformer backbone, language modeling tasks, and token-based processing), but VLMs include specialized components to handle visual input, such as an image encoder and visual tokens. For instance, the VL–Rethinker [9] framework enhances multimodal "slow-thinking" by explicitly enforcing a rethinking phase during inference, encouraging models to assess and correct their own predictions. Similarly, the R^3V (Reflect, Revise, and Re-select Vision) [10] paradigm enables VLMs to iteratively generate, critique, and revise interpretations at test time, yielding substantial gains in reasoning benchmarks. Additionally, self-feedback mechanisms have been shown to improve semantic grounding the alignment between image regions and generated text highlighting a promising direction for mitigating hallucinations and misinterpretations [11]. These advances indicate that encouraging self-verification in VLMs may enhance their reliability and robustness in tasks like contextual emotion recognition.

In this paper, we aim to further investigate the contextual biases in Vision–Language Models and their impact on emotion recognition. Our hypothesis suggests that subtle alterations in background context—while preserving the same body pose—can significantly influence VLM interpretations of emotional states. To explore this, we altered existing datasets, utilizing images from the BEAST [12] and EMOTIC [4] datasets. We extracted individual subjects using YOLOv8 and synthetically placed them onto diverse, unrelated background scenes selected from the Landscape Pictures dataset (https://www.kaggle.com/datasets/arnaud58/landscape-pictures, accessed on 12 May 2025), while maintaining their original pose and expression. We then analyzed how VLM-generated descriptions and emotion predictions changed across these varied contexts.

Our experimental results demonstrate a strong contextual effect: in over 90% of the cases analyzed, predictions for the same body posture differed based on the background, highlighting the VLM's susceptibility to contextual influence. To reduce this bias, we propose a multimodal approach that integrates four streams: (1) the original image, (2) the isolated body posture, (3) the automatically generated scene description, and (4) a specifically prompted "discrepancy description." The discrepancy description is generated by prompting the VLM to explicitly describe the mismatch if any between the emotion displayed by the person and the atmosphere suggested by the background. This encourages the model to reflect critically on whether the context aligns with the subject's emotion.

By fusing these diverse inputs, our approach moves toward a more robust and context-independent understanding of human emotions.

The rest of the paper is organized as follows: Section 2 reviews related work in emotion recognition and Vision–Language Models. Section 3 describes the datasets used in our study. Section 4 outlines the experimental methodology for isolating VLM bias in contextual emotion recognition, while Section 5 introduces our proposed multimodal approach. Section 6 presents the results, followed by a discussion in Section 7. Finally, Section 8 concludes the paper.

2. Related Work

Emotion is a complex psychological state involving three key components: a subjective experience, a physiological response, and a behavioral or expressive reaction [13]. While the first two are primarily studied in psychological and neurological research, the expressive component has been the focus of most computer vision-based emotion recognition systems. Despite substantial progress, several limitations persist, such as poor generalization across cultures, over-reliance on facial expressions, and insufficient handling of ambiguous or multimodal cues.

Recent efforts aim to bridge insights from psychology and machine perception. One key insight is that relying solely on visual analysis of facial expressions or body posture is often suboptimal. Consequently, alternative modalities such as audio, textual description, or scene context are being explored to complement visual cues. These additional streams, including context-aware captioning or pose-based analysis, introduce a richer multimodal understanding of emotion.

2.1. Beyond Emotion Recognition from Face Expression

Analyzing body posture is part of the behavioral or expressive aspect of emotions. Although relatively new and with significantly less research compared to facial expression analysis, body posture-based emotion inference has seen some noteworthy prior studies [14]. Into this direction, Karg et al. [15] conducted a comprehensive survey highlighting the use of motion capture data to derive affective expressions from body movements. By applying clustering techniques to these dynamic features, they successfully categorized emotions into distinct classes, corresponding to different regions of the valence–arousal space, such as joy, anger, sadness, and fear. Tu et al. [16] explored skeleton-based human action recognition using spatial–temporal 3D convolutional neural networks (CNNs). They applied these techniques to emotion recognition tasks, identifying key emotional states such as anger, joy, and sadness from skeletal geometric features. Their results demonstrated that advanced CNN architectures could effectively capture the temporal dynamics of body movements, leading to improved emotion classification accuracy.

Moving away from unimodal data, Zhang et al. [17] provided an extensive review of emotion recognition techniques that leverage multimodal data. Their work emphasized the integration of various data sources, including facial expressions, body gestures, and speech,

to improve the accuracy of emotion classification. Through a detailed analysis of different machine learning methods, they concluded that multimodal approaches significantly enhance the performance of emotion recognition systems, particularly in distinguishing complex emotional states. Yang and Narayanan [18] analyzed the interplay between speech and body gestures under different emotional conditions. Their research highlighted the influence of emotional states on the synchronization of speech and gestures, showing that this interplay could be effectively modeled using prosodic and motion features to improve emotion recognition performance. Poria et al. [19] focused on multimodal sentiment analysis by integrating multiple levels of attention mechanisms across different modalities, including text, audio, and visual data. Their study revealed that combining these modalities using deep learning frameworks significantly improves the system's ability to detect and classify emotions accurately, particularly in complex, real-world scenarios. Mittal et al. [20] introduced a context-aware multimodal emotion recognition system, named EmotiCon, which integrates various contextual cues alongside traditional emotion recognition features. By applying this approach, they successfully recognized a wide range of emotions in different environments, demonstrating the importance of contextual information in enhancing the robustness of emotion recognition systems.

Even further, Ranganathan et al. [21] explored multimodal emotion recognition using deep learning architectures that combine facial, body, and speech modalities. Their findings showed that using deep learning models to integrate these modalities leads to substantial improvements in the accuracy of emotion detection, particularly in noisy and challenging environments. Psaltis et al. [22] employed a late fusion strategy: they successfully combined facial action units and high-level body gesture representations to recognize emotions such as surprise, happiness, anger, and sadness. Aqdus et al. [23] proposed a deep learning approach for emotion recognition that combines upper-body movements and facial expressions. Their model demonstrated superior performance in recognizing emotions such as happiness, sadness, and anger, particularly when using feature-level fusion of body and facial cues. This bimodal approach underscored the advantages of integrating multiple expressive channels in emotion recognition tasks.

These works collectively suggest that multimodal systems incorporating context and pose information provide significant performance gains compared to face-only models. However, the field still struggles with challenges such as contextual ambiguity, cultural variability, and overfitting to scene-specific biases.

2.2. Language Models in Emotion Recognition

Parallel to these developments, the rise of Large Language Models (LLMs) has opened new avenues for emotion analysis—particularly in text-based empathy and sentiment understanding. Buscemi et al. [5] evaluate the performance of four LLMs—ChatGPT 3.5, ChatGPT 4, Gemini Pro, and LLaMA2 7B—on sentiment analysis tasks across ten languages. The goal is to assess how well these models handle ironic, sarcastic, or culturally ambiguous sentiments. To this end, the researchers designed 20 complex sentiment scenarios, translated into 10 languages, and asked each model to rate sentiment on a scale from 1 (unhappy) to 10 (happy), comparing the outputs with human ratings. Both ChatGPT models showed strong alignment with human judgments but struggled with irony and sarcasm. Gemini Pro demonstrated better sarcasm detection but tended to produce more negative sentiment scores overall. LLaMA2 7B consistently gave more positive ratings but lacked nuanced interpretation. The study highlights language-specific biases and the need for greater transparency and bias mitigation in model training.

In a related study, Welivita et al. [6] examined the empathic capabilities of state-of-the-art LLMs compared to human responses. Focusing on 32 fine-grained positive and negative

emotions, they used human evaluation as the primary metric. Drawing on 2000 prompts from the EmpatheticDialogues dataset, they enlisted 1000 human evaluators, divided into five groups, each assessing either human-generated or LLM-generated responses. Results showed that LLMs often provided more emotionally validating and supportive responses than humans.

Extending this line of inquiry to visual data, Bhattacharyya et al. [24] conducted a comprehensive evaluation of Large Vision–Language Models (VLMs) in emotion recognition from images, addressing the lack of detailed assessments in this area. The authors explored factors influencing recognition performance and analyzed various error types. They assessed VLMs' ability to recognize evoked emotions from image prompts, their robustness to prompt variations, and the nature of errors made. Using existing emotion datasets, they introduced the Evoked Emotion benchmark (EVE), featuring a diverse set of image types and challenging samples. Their findings reveal that current VLMs struggle with emotion recognition, particularly when prompts lack explicit class labels. The presence of such labels significantly improves performance, while prompting VLMs to adopt emotional personas tends to degrade results. A human evaluation further confirmed that performance limitations stem from both the models' intrinsic capabilities and the inherent complexity of the task.

In the current state of the art on emotion recognition benchmarks, Xenos et al. [25] propose a novel approach that emphasizes the critical role of contextual information in affective computing. Their two-stage method leverages the generative capabilities of VLMs. In the first stage, the LLaVA model generates natural language descriptions of a subject's emotional state in relation to the visual context. In the second stage, a Transformer-based architecture fuses visual and textual features for final emotion classification. Experiments on EMOTIC, CAER-S, and BoLD datasets demonstrate that this context-aware approach outperforms unimodal methods and achieves state-of-the-art results, underscoring the importance of contextual cues in emotion recognition.

3. Datasets

In this work, we use data from two existing datasets that are publicly available for studying emotion from context and body posture. These are EMOTIC and BEAST.

3.1. EMOTIC

The EMOTIC dataset [20] is a comprehensive collection of images depicting people in various unconstrained environments, annotated based on their apparent emotional states. This dataset includes 23,571 images and 34,320 annotated individuals with 66% being male and 34% female. Among these, 10% are children, 7% are teenagers, and 83% are adults. Images were either manually gathered via Google using queries about places, social settings, activities, and emotions or from COCO and Ade20k. Thus, diverse contexts of people in various environments and activities have been ensured.

The images were annotated using Amazon Mechanical Turk (AMT), where annotators were able to make reasonable guess about the subjects emotional states. The 26 class labels are Affection, Anger, Annoyance, Anticipation, Aversion, Confidence, Disapproval, Disconnection, Disquietment, Doubt/Confusion, Embarrassment, Engagement, Esteem, Excitement, Fatigue, Fear, Happiness, Pain, Peace, Pleasure, Sadness, Sensitivity, Suffering, Surprise, Sympathy, and Yearning. This list of 26 emotion categories was created considering the Visual Separability criterion: words with close meanings that are not visually separable were grouped together. For instance, Anger includes words like rage, furious, and resentful. Although these affective states differ, they are not always visually separable

in a single image. Thus, the list of affective categories can be seen as the first level of a hierarchy, with each category having associated subcategories.

After the annotations (one annotator per image), the images were divided into three sets: Training (70%), Validation (10%), and Testing (20%), maintaining a similar distribution of affective categories across the sets. Additional annotations were collected: four more for the Validation set and two more for the Testing set, resulting in a total of five annotations for the Validation set and three for the Testing set, with slight variations for some images due to the removal of noisy annotations.

In Figure 1, we can see the number of annotated individuals for each of the 26 emotion categories sorted in descending order, according to the authors of the dataset. The dataset is highly unbalanced, presenting a particular challenge. There are more examples for categories linked to positive emotions, such as Happiness or Pleasure, while those associated with negative emotions, like Pain or Embarrassment, have very few examples. The category with the most examples is Engagement, as most images depict people involved in some activity, indicating some level of engagement.

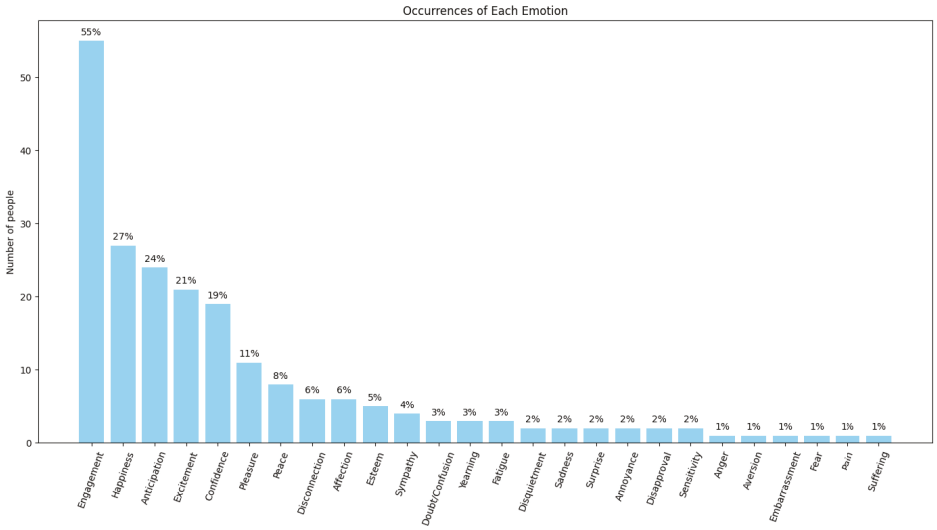


Figure 1. Distribution of annotated individuals per emotion category in the EMOTIC dataset, highlighting the imbalance across classes.

The database creators also analyzed the co-occurrence rates of various emotion categories within the dataset to understand how frequently one emotion appears in conjunction with another. This analysis employed clustering techniques such as K-Means to identify groups of emotions that commonly co-occur. The identified clusters include the following:

- Anticipation, Engagement, Confidence;
- Affection, Happiness, Pleasure;
- Doubt/Confusion, Disapproval, Annoyance;
- Yearning, Annoyance, Disquietment.

These clusters reveal a significant challenge for emotion recognition algorithms. Emotions within these clusters often exhibit overlapping features or contexts, making it difficult for algorithms to distinguish between them accurately. For example, both Anticipation and Confidence may involve positive expectations, which could manifest similarly in body language, complicating differentiation. Similarly, Affection, Happiness, and Pleasure all convey positive sentiments, potentially leading to confusion in automated systems trying to parse these subtle distinctions.

Furthermore, emotions like Doubt/Confusion and Disapproval can both display signs of discomfort or hesitation, adding to the complexity of their separation in emotion recog-

nition systems. The overlap in emotional expressions and the variability in contextual cues can lead to misclassification or ambiguous interpretations by algorithms. This complexity highlights the need for sophisticated methods and careful tuning in emotion recognition technologies to accurately capture and distinguish these nuanced emotional states.

Figure 2 displays a selection of EMOTIC examples across distinct emotion categories. These demonstrate the dataset's diversity in environmental context, body language, and visual appearance.



(**a**) Labels: Affection, Anticipation, Engagement, Esteem, Excitement, Happiness, Pleasure



(**b**) Labels: Confidence, Engagement, Excitement, Fatigue



(**c**) Labels: Happiness, Pleasure



(**d**) Labels: Excitement

Figure 2. Sample images from the EMOTIC dataset showcasing different emotional labels and environments. Each image reflects one or more of the dataset's 26 emotion categories.

3.2. BEAST

The BEAST database [12] contains 254 whole-body expressions from 46 actors, each expressing one of four emotions: anger, fear, happiness, and sadness. Actors were individually instructed to display the emotions using their whole body, following a standardized procedure. Each emotion was associated with specific, representative daily scenarios: anger involved being in a quarrel and threatening retaliation, fear involved being pursued by an attacker, happiness involved meeting an old friend, and sadness involved learning of a dear friend's passing. The actors' faces were blurred in all images and participants were asked to categorize the emotions in a four-alternative forced-choice task. The dataset serves as a valuable tool for assessing recognition of affective signals, useful in explicit recognition tasks, matching tasks and implicit assessments.

Emotions like disgust were not included as a pilot study, which may indicate these were difficult to recognize without facial cues [12]. Based on the photographer's assessment of posture expressiveness, multiple pictures of the same actor and emotion were sometimes taken to enrich the dataset with slight variations. The total number of pictures was 254 (64 anger, 67 fear, 61 happiness, and 62 sadness). The scenarios were presented in random order to each actor. After the photo shoot, the images were desaturated and the facial areas blurred. Examples of the pictures can be seen in Figure 3.



Figure 3. Example images from BEAST dataset. We illustrate one example for each class.

A high level of consistency has been reported among the dataset raters, using the Fleiss' Generalized kappa metric and obtaining a value of 0.839. Also, the overall accuracy of the raters was 92.5%.

Authors concluded that all emotions were well recognized. Sadness was the easiest to identify, followed by fear, while happiness was the most difficult. These findings align with those of Atkinson et al. [26], suggesting inherent differences in the recognizability of bodily emotions, although different tasks may influence these results. Incorrect categorizations revealed that anger was more often confused with happiness than sadness and fear was more frequently confused with anger and happiness than sadness, this pattern contrasts with facial expression studies where happiness is usually better recognized than negative emotions. All emotions were expressed voluntarily, which may differ from natural expressions, even though real-life scenarios were used to enhance validity. There were no significant inter-individual differences in emotion categorization accuracy, indicating varied recognition abilities among subjects.

The two databases share some similarities (both have been built to express emotion by body posture) but they also have critical differences:

- BEAST is small. EMOTIC is larger.
- BEAST illustrates prototypes for body emotions. EMOTIC illustrates various and diverse examples for each emotional body posture.
- BEAST is devoid by background. EMOTIC sought background context that enhances person emotion.
- BEAST has removed face expression. EMOTIC has face expression.

The selection of EMOTIC and BEAST was motivated by their complementary characteristics and distinct advantages. EMOTIC offers a large-scale, diverse collection of emotional expressions captured in natural, unconstrained environments, incorporating both facial and bodily cues along with rich contextual backgrounds. In contrast, BEAST provides controlled, prototypical whole-body expressions with facial cues intentionally removed, making it particularly useful for isolating the contribution of body posture. Together, these datasets enable a more comprehensive exploration of both naturalistic and standardized representations of emotion.

Despite its scale and diversity, EMOTIC presents several limitations. The dataset exhibits a skewed gender distribution (66% male), and underrepresentation of younger age groups such as children and teenagers. It is also highly imbalanced, with some emotion categories comprising less than 2% of the data. Moreover, many images only show partial body views, typically from the elbows upward, limiting the available posture

information. These constraints can introduce biases and reduce the generalizability of emotion recognition models trained solely on this dataset, highlighting the importance of critical evaluation when interpreting results.

BEAST, on the other hand, is a much smaller dataset with only 254 images, which constrains the training of data-hungry models and limits variability across subjects and expressions. Nonetheless, BEAST offers several advantages: the use of standardized scenarios enhances consistency across emotional expressions, the balance between the four classes, and the intentional blurring of faces ensuring that emotion must be inferred solely from body posture. This makes it particularly well-suited for evaluating the body's contribution to emotion recognition in isolation.

4. Methodology

We formulate the hypothesis that current state-of-the-art solutions in body emotion recognition are over-reliant on the context. These solutions assume that **person emotion is highly related to image background (context)**.

To test this hypothesis, we evaluate a leading Vision-and-Language Model (VLM)-based emotion recognition system on a modified dataset where expressive full-body figures from the BEAST dataset are placed against misleading backgrounds. The goal is to decouple body posture from its natural scene and observe the model's reliance on context. Our preliminary findings show a significant performance drop; classification accuracy falls from 90% to 55%, highlighting the model's dependence on contextual cues rather than body posture alone.

To build toward this result, we begin by reviewing the key principles of context-based emotion recognition and the architecture of a representative VLM model. We then describe the experimental setup used to validate our hypothesis.

This section describes the effort to isolate and diagnose problems of previous approaches. Following the diagnosis and identification of their causes, the next section, entitled "Multimodal Approach", presents the proposed solution, which is designed to add targeted fixes without disrupting what already works.

4.1. Standard Context-Based Emotion Recognition

Currently state-of-the-art performance in recognizing emotions in context is achieved by complex solutions with an intricate architecture. We detail by referring specifically to the work of Xenos et al. [25]. A better view of the architecture can be seen in Figure 4. The prototypical solution has a two-stage framework for applying Vision-and-Language Models (VLMs), e.g., LLaVA v1.5, to enhance emotion perception in context-heavy scenes.

During Stage I, the VLM generates natural-language description of each person's emotional experience towards their environment. Such "emotion-aware" description (e.g., "the person inside the red box is fearful because they are backing away from an oncoming car") encapsulates how context-informed information redefines emotional comprehension.

In the second stage, the text description is represented by a RoBERTa model [27] and the associated image is represented by a visual encoder (e.g., ViT-L). Such generated language and vision embeddings are combined by using a Q-Former module [3] based on cross-attention to merge context semantic information and visual information. The combined representation is input to a fully connected layer for final emotion classification derivation.

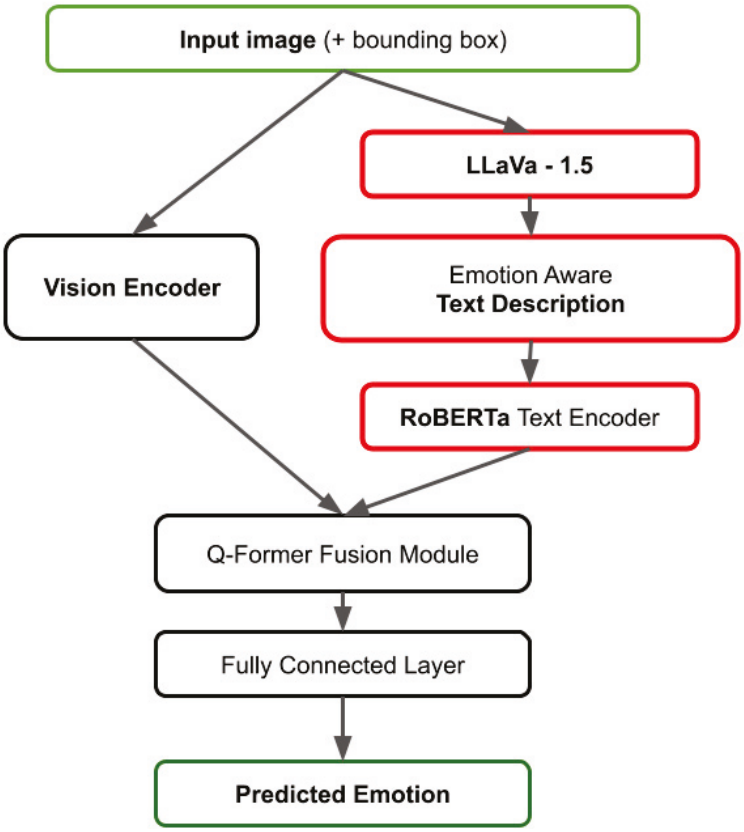


Figure 4. Architecture diagram of state-of-the-art contextual emotion recognition solution. The solution uses parallel branches where embedding computed from only visual data and visual-to-text-to-embedding, respectively, are combined for the final descriptor. The branch marked by red boxes is shown to be the key contributor to the overall performance.

One of the main advantages of this approach is its ability to leverage common sense reasoning. Vision–Language Models rely on pre-trained knowledge, allowing them to understand and interpret complex physical and social scenarios without the need for manually crafted rules. This method generates descriptive text at inference time, removing the necessity for manual context labeling. Another benefit is enhanced interpretability by producing textual explanations for emotion predictions. Finally, the integration of both contextual text and visual input through multimodal fusion improves the system's robustness, especially in cases where facial expressions or body language alone might be too ambiguous to yield accurate emotional interpretation.

A similar setup—featuring a wide range of alternative solutions at each step but focusing on Vision-and-Language Models (VLMs) and Large Language Model (LLM) descriptors—is explored by Etesam et al. [28]; based on results with alternative VLM and LLM models for the respective blocks, while differences appear, they are minor compared to the situation when they are not used.

In Figure 4, we point the attention to the blocks marked with red that are main contributors to the model's success, as emphasized by Xenos et al. [25]. While evaluated in benchmarks such as EMOTIC, these blocks are net contributors; more precisely, they improve the mAP on the EMOTIC database by approximately 12% compared to solutions that do not use them. These results are consistent with those reported by Etesam et al. [28], who varied the VLM model. However, we argue that the overall solution is over-reliant on these blocks, introducing a certain bias toward the background.

To emphasize the background over-reliance, we test the solution on images from BEAST merged with uncorrelated background. Examples in this direction, with the corresponding prediction, may be seen in Figure 5. One might note incorrect predictions.

These observations underline the importance of testing such context-aware models on data that deliberately decouple body posture from its surrounding scene. By combining emotion-rich body expressions from BEAST with irrelevant or misleading backgrounds, we can assess the true extent of context dependence, confirming that current models tend to overfit to contextual cues rather than bodily expressions alone.



(**a**) Label: Angry - Pred: Fearful (**b**) Label: Sad - Pred: Angry (**c**) Label: Fearful - Pred: Angry (**d**) Label: Happy - Pred: Angry

Figure 5. Prediction examples on images obtained by over-imposing persons from BEAST database with some uncorrelated background, using the previous state-of-the-art architecture.

4.2. VLM Experiments

Motivated by the previous findings, in this subsection, we design a series of experiments to better understand the role of context and context captioning in emotion recognition. These experiments are synthesized in Figure 6 and detailed in the next paragraphs. All experiments focus exclusively on Vision–Language Models (VLMs).

Among the various available VLMs, we primarily use Gemini, as it imposes fewer limitations on the number of experiments compared to other alternatives.

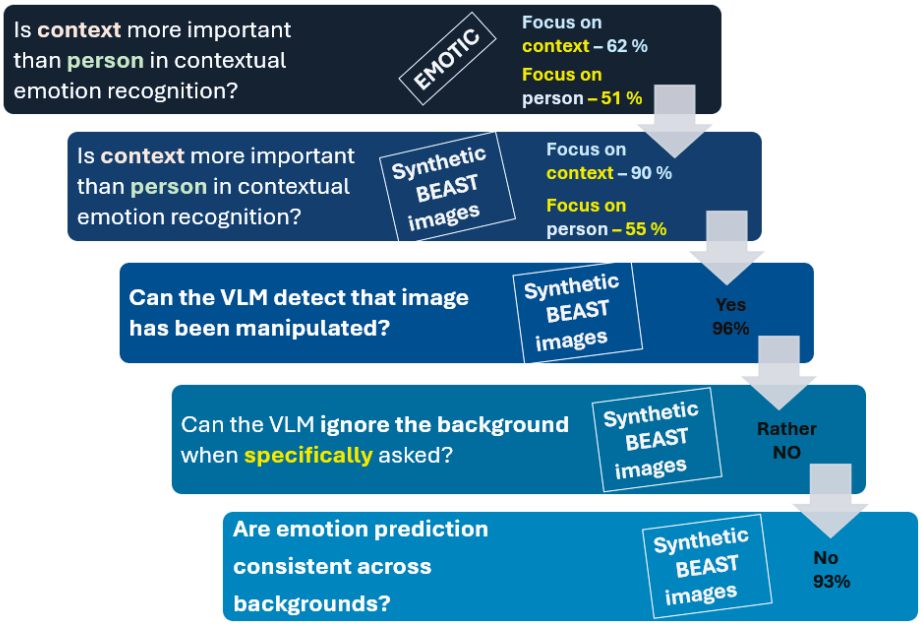


Figure 6. The chain of hypothesis verified for VLM reliance on background in captioning context in emotion. Text with different color marks key detail in the question or database used.

4.2.1. Context Weight in Emotion Recognition

We investigated the role of context in emotion recognition by comparing the model's predictions when analyzing the full image versus only the segmented person. First, we sampled 300 images from EMOTIC, where a full-body pose can be visible. In these images, we asked the VLM Gemini to predict the emotion. We found that in 51.13%, when the analysis is focused on the person, the model correctly predicted the emotion, compared to 62.05% of cases when viewing the full image (i.e., the person is in a context). This indicates that background context significantly influences the model's emotion interpretation. Examples of images, prompts, and VLM's reported captions are shown in Figure 7.



Figure 7. Alternative interrogation of VLM over images from EMOTIC database: prompt, image, and results for the situation when the captioning was required to be only on the body (Case 1) and, respectively, on the image context (Case 2).

This result is in line with findings from [25], which reported an increase in performance from 25.55 when only the image content from person bounding box was used as input in the Visual Transformer (ViT) model, to 34.25 when a VLM was asked to recognize the emotion, and to 38.52 when the solution included emotion awareness and context caption was used.

To further examine this dependency, we also tested on BEAST in two scenarios. To evaluate the formulated hypothesis, we first segmented the individuals from the BEAST dataset and placed them onto various backgrounds, creating synthesized versions of the original images. The first scenario is when the background was in correlation with the expressed emotion while the second assumed that person emotion and background were in contradiction. We tested these hypotheses by training the RoBERTa classifier with the given descriptions from Gemini.

For the first scenario, description accuracy led to a performance of 90%, while for the contradictory scenario, it dropped down to 55.13%. This result highlights the model's over-reliance on context, which may lead to errors when using simple prompts.

4.2.2. Can the VLM Detect Unnatural Correlation?

The experiments demonstrating an over-reliance on background are based on synthetically created composite images. A natural concern is whether such images are too simplistic or idealized for meaningful evaluation. To address this, we aimed to verify whether Vision-and-Language Models (VLMs) can detect the artificial nature of these altered images.

Prior work by Wang et al. [9] showed that VLMs possess some ability to assess the consistency and authenticity of visual–linguistic content. Motivated by this, we tested Gemini 2.0 [29] to determine if it could distinguish between original and manipulated images.

The test images were constructed by extracting people from the BEAST and EMOTIC datasets using the YOLOv8 object detection model and compositing them onto random scenes from the Landscapes dataset. This process creates realistic yet contextually mismatched visuals, simulating real-world image manipulation more faithfully than simple cut-and-paste examples.

We submitted each composited image with the prompt: *"Please determine if the image is original (photograph) or modified (synthesized)."* Gemini 2.0 correctly classified 96% of the images as "modified," indicating a strong ability to detect such manipulations even in relatively complex, realistic composites. While this high performance is promising, we acknowledge that altered images can vary widely in realism and style, and further testing across more diverse manipulation types would be needed to generalize this finding fully.

4.2.3. Refined Prompt

Based on the previous found conclusions, we modified the prompt to a more refined one: *"Based on this emotion list ['angry','fearful','happy','sad'], please analyze the emotions expressed by the person in the image. If you identify that the image is synthetic or edited, please focus only on the person and ignore the background."*. This instruction led to a slight increase in the model's accuracy, from 55.13% to 63.72%, suggesting that prompting can *partially* mitigate context bias.

Even with explicit instructions to focus on the subject and ignore the background, visual context continues to influence the model's interpretation. To further illustrate this phenomenon, we present two concrete examples in Figure 8, where the same body pose is interpreted differently depending on the background or the subjective ambiguity of the scene.

In Figure 8a, the model confidently associates the pose with happiness, likely influenced by either the openness of the posture or subtle contextual elements in the image. In contrast, Figure 8b receives a more uncertain, context-agnostic interpretation. Although the model follows the prompt by acknowledging the synthetic nature of the image and focusing on the pose, it still struggles to confidently classify the emotion, suggesting fear or surprise as possibilities.



(**a**) (**b**)

Figure 8. Example of context-sensitive emotion predictions: (**a**) "The person in the image appears to be expressing happiness." (**b**) "The image looks edited, so I'll focus on the person. The individual's posture and outstretched arms could suggest a sense of fear or surprise. It's difficult to say definitively without a visible face".

4.2.4. Consistency

To further probe the behavior of Vision–Language Models, we evaluated their consistency in emotion predictions across varying visual contexts. Specifically, we examined whether the model could maintain stable emotion predictions for the same body pose when composited onto different backgrounds. Figure 9 illustrates this setup, showing a single body pose (annotated as "Happy") placed in five distinct scenes, each varying in environment, lighting, and semantic content.



Figure 9. Same body pose (marked as Happy) in five different background contexts.

Analysis Method: Each prediction was grouped by body pose identity, resulting 254 sets of 5 images. We then examined whether the model predicted the same emotion for a given pose across all five backgrounds.

Initial experiment with a simple prompt:

- Only 18 out of 254 poses (7%) received consistent emotion predictions across all five backgrounds.
- The remaining 236 poses (93%) yielded varying predictions depending on the background.

Thus, it further enhanced the perception of over-reliance on background when suggesting the emotion of a person shown in full body.

Second experiment with a refined prompt that specifically asks to "Ignore background if the image is edited":

To move further, we aimed to precisely instruct the VLM; give him a "shorter leash". The prompt was adjusted to explicitly instruct the model to ignore background context when detecting edited images:

- A total of 26 out of 254 poses (10%) produced consistent predictions across all backgrounds.
- A total of 228 poses (90%) still showed variability, although the consistency slightly improved.

While the 3% increase indicates a slight improvement, it likely reflects natural variation rather than a true enhancement in prediction reliability. This reinforces the concern that background changes continue to influence the model's outputs.

Even with explicit instructions to ignore the background, the same body pose still triggered inconsistent predictions. This underscores our broader finding: environmental context frequently overshadows bodily cues in emotion interpretation. Despite refined prompting, Large Vision–Language Models remain sensitive to background signals, leading to unstable or ambiguous outcomes. Such behavior shows the need for multimodal grounding as pursued in our fusion architecture. To anchor predictions in structured, complementary information, we add another branch to the solution, which encodes explicitly the pose of the person in the image. This solution will be further detailed and evaluated in Section 5.

4.3. Completely Synthetic Images

Finally, we conducted an experiment involving deliberately contradictory image prompts. Using DALL-E [30,31], we generated 15 synthetic scenarios in which the emotional body language conflicted with the surrounding scene, such as

- "Man in a defensive posture at a wedding;"
- "Crying child at a birthday party;"
- "Ecstatic woman at a funeral;"
- "Joyful child in a war zone;"
- "Panicked man at a meditation retreat."

Figure 10 presents several of these generated images. Visually, these examples feature strong emotional expressions on the central figure while embedding them in semantically rich and emotionally incongruent backgrounds.

We then asked the model to describe these images and it generated accurate emotion descriptions in 100% of cases. In contrast to synthetic examples created by blending content from two existing images, DALL-E-generated images are typically dominated by the foreground person, whose facial expression is at its most expressive. The results on these images may suggest that modern Vision–Language Models prioritize facial expression as the main feature for emotion recognition.

Concluding these experiments on VLM while analyzing emotion in context:

1. When available, VLMs give the highest priority to the face. This aligns with human behavior, but recent perspectives criticize this approach as intrusive, since the face is closely tied to personal identity.

2. VLMs give secondary priority to the background, which is a weakness, as the context inferred from the background is merely circumstantial in relation to emotion.
3. Only afterward is body posture taken into account.
4. VLMs are capable of identifying unnatural compositions. However, they are not inherently skeptical, and when tasked with emotion recognition in context, they revert to their default prioritization.



Prompt: "Defensive man at a wedding"



Prompt: "Happy woman at a funeral"

Figure 10. Examples of AI-generated (DALL-E) images based on contextually contradictory prompts that are shown below each figure.

5. Multimodal Approach

To address the context sensitivity identified in state-of-the-art VLMs, we propose a multimodal learning architecture named BECKI (Body Emotion Context Keypoint Image) that explicitly integrates information from several distinct sources: body keypoints, raw grayscale body images, contextual descriptions, and emotion-focused textual cues. The goal is to ensure the model's emotion recognition capabilities are more robust and less prone to background or context manipulation. While in the previous section, we have presented a series of experiments designed to isolate the contextual bias issue, in this section, we propose a solution that is designed to add targeted fixes without disrupting the functional parts.

The core dataset used for benchmarking is derived from the BEAST database, which provides grayscale full-body images labeled with four basic emotions: angry, fearful, happy, and sad. Although the dataset is relatively small and limited in subject diversity, its design enables controlled manipulation of the background and offers a clear view of body pose, making it particularly suitable for evaluating sensitivity to contextual bias. To this end, we replaced the original neutral backgrounds with environments that contradict the expressed emotion. Each modified sample was further enriched with two levels of semantic annotation: a contextual description of the scene and a specifically constructed "emotion analysis" description, designed to highlight potential discrepancies between the emotion expressed and the surrounding environment.

Our approach builds on a custom neural pipeline, graphical represented in Figure 11. Compared to previous approaches, exemplified in Figure 4, the main innovation lies in the addition of blocks outlined in red; also, the findings with respect to refined prompt concluded in the necessity of dividing the captioning branch in two, which is outlined by blocks in blue. Next, we will detail the key blocks.

Given an image, by adding two captions and a set of posture keypoints, it is formed into a data point. More precisely, a data point consists of

1. Grayscale image of the subject;
2. Structured file of 3D keypoints from body pose estimation;
3. Contextual textual description;
4. Emotion-focused textual analysis.

These data points have been gathered into a so-called "MultimodalBEASTDataset". To work with all these modalities we have developed a specialized model that incorporates the following:

- **MobileNetV2 CNN**: Serves here primarily to extract local appearance cues without dominating the model's focus since global dependencies are already captured via text and keypoint streams.
- **KeypointEmotionTransformer**: A lightweight Transformer encoder specifically designed for structured human pose data. Crucially, keypoints are normalized relative to image size to reduce scale and location variability.
- **Text Encoders (RoBERTa-base)**: We use two parallel RoBERTa-base models, one for the contextual scene description and one for the emotion-focused analysis. This dual stream approach allows the model to separately process general and emotionally salient text, which is especially useful in ambiguous or contradictory background situations. Both models are fine-tuned on task data.
- **Fusion and Classification**: The outputs of all four branches (image, keypoints, context text, emotion text) are concatenated into a unified feature vector. This vector is then passed through a two-layer MLP: a linear projection to 256 dimensions, ReLU activation, dropout of 0.3, and a final classification layer producing logits over the four emotion classes.

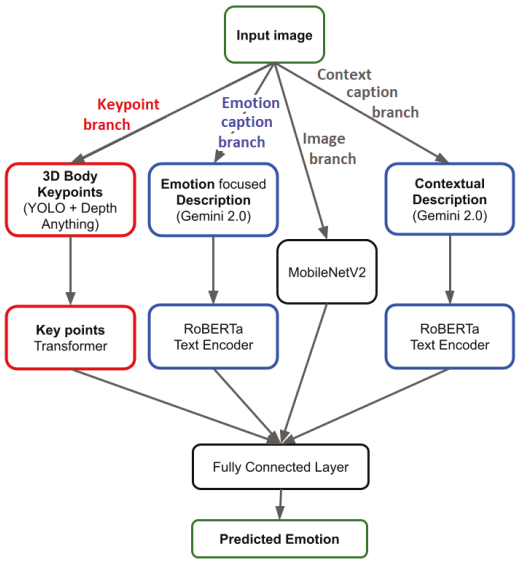


Figure 11. Overview of the multimodal emotion recognition pipeline (named BECKI) integrating image, keypoint, and textual modalities. Compared to the previous solution, here, embeddings from three branches are gathered. The red blocks mark our key proposal.

5.1. Keypoint Transformer Architecture

To enhance subject-focused emotion recognition and reduce reliance on background context, we introduce the **Keypoint Transformer**. It is a lightweight Transformer-based architecture specifically designed for structured human pose data. Each input sample consists of 17 keypoints, where each keypoint is described by a 4-dimensional vector: 2D coordinates (x, y), confidence score c, and depth d.

5.1.1. Embedding and Positional Encoding

Each keypoint vector $\mathbf{k}_i \in \mathbb{R}^4$ is first mapped to a higher-dimensional representation via a linear embedding layer:

$$\mathbf{e}_i = W_e \mathbf{k}_i + \mathbf{b}_e, \quad \mathbf{e}_i \in \mathbb{R}^{128} \tag{1}$$

Here, $W_e \in \mathbb{R}^{128\times 4}$ and $\mathbf{b}_e \in \mathbb{R}^{128}$ are learnable parameters. To retain structural ordering of the keypoints, we add a learnable positional encoding matrix $\mathbf{P} \in \mathbb{R}^{17\times 128}$:

$$\mathbf{z}_i = \mathbf{e}_i + \mathbf{P}_i \tag{2}$$

The resulting sequence $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{17}] \in \mathbb{R}^{17\times 128}$ forms the input to the Transformer.

5.1.2. Transformer Encoder

The encoder follows the standard architecture of Vaswani et al. [32], but is adapted for compact structured input. We use a stack of three Transformer encoder layers, each with four attention heads, a hidden dimension of 128, and a feedforward dimension of 256. Dropout with rate 0.1 is applied within each layer. The encoder is implemented using PyTorch's default post-normalization design.

$$\mathbf{H} = \text{TransformerEncoder}(\mathbf{Z}) \in \mathbb{R}^{17\times 128} \tag{3}$$

5.1.3. Global Aggregation and Classification

The output sequence is reduced to a global representation via adaptive average pooling across the keypoint dimension:

$$\mathbf{h}_{\text{global}} = \text{AvgPool1D}(\mathbf{H}) \in \mathbb{R}^{128} \tag{4}$$

Finally, a fully connected classification layer produces logits for each of the target emotion classes:

$$\mathbf{o} = \mathbf{W}_{\text{fc}} \mathbf{h}_{\text{global}} + \mathbf{b}_{\text{fc}}, \quad \mathbf{o} \in \mathbb{R}^C \tag{5}$$

In our experiments, we set $C = 4$, corresponding to emotion categories such as happy, sad, angry, and neutral.

The hyperparameters of the Transformer encoder—three layers, four attention heads, 128 hidden dimensions, and 256 feedforward dimensions—were selected based on empirical evaluation across multiple configurations. We experimented with varying the number of layers (1–6), the width of the model (64–256), and the heads of attention (2–8) and found that this configuration offers the best trade-off between the accuracy of the classification and the size of the model. Our goal was to maintain a lightweight architecture suitable for emotion recognition from keypoints, without overfitting to the relatively compact input structure of 17 tokens.

5.2. Textual and Visual Backbone Models

5.2.1. RoBERTa for Textual Encoding

For the textual modalities—contextual descriptions and emotion-focused analysis—we utilize two parallel RoBERTa encoders [27]. RoBERTa is a robust, pre-trained Transformer-based language model built on BERT architecture, which has been extensively fine-tuned on large amounts of data for better language understanding and representation. Its ability to capture nuanced semantic information makes it well-suited to extract meaningful features from complex, context-dependent textual inputs.

We recall that RoBERTa uses a Byte-Pair Encoding (BPE) tokenizer, which first it breaks the input text into subword units and converts each token into a unique integer ID using a predefined vocabulary. Next, the token IDs are passed, consecutively, through the token embeddings' look-up table that maps token into vectors, and positional embeddings, which add information about token positions (i.e., adds sequence awareness). The embeddings are summed and fed into the encoder, which is a stack of Transformer encoder layers [27]. The final output is a sequence of embeddings, one for each input token.

The proposed solution uses two independent RoBERTa encoders, allowing the model to learn distinct representations for general context versus emotion-specific textual cues, which is critical for disentangling subtle emotional signals from the surrounding environment.

5.2.2. MobileNetV2 for Visual Encoding

The image modality is encoded using MobileNetV2 [33], a lightweight convolutional neural network architecture designed for efficient mobile and embedded vision applications. MobileNetV2 leverages depthwise separable convolutions and inverted residual blocks to significantly reduce computational cost and model size while maintaining competitive accuracy. Although other CNN architectures (e.g., ResNet, EfficientNet) or Vision Transformers (ViTs) could be used, MobileNetV2 offers an advantageous trade-off between performance and computational efficiency, enabling faster training and inference, particularly beneficial in a multimodal setup where multiple encoders run in parallel. Its adaptability to grayscale images further supports the focus on body appearance without reliance on color information.

This combination of specialized Transformer-based text encoders and a lightweight CNN backbone provides a balanced approach to extracting rich, complementary representations from different data sources while keeping the overall model size and complexity manageable.

Training Details

The fusion model was trained using the AdamW optimizer with a fixed learning rate of 1×10^{-5} and weight decay for regularization. We employed a batch size of 16 and trained for up to 50 epochs. To address class imbalance (in case of EMOTIC), we applied weighted cross-entropy loss with class weights computed from the training label distribution. Although no explicit learning rate scheduler or warm-up strategy was used, we monitored validation performance after each epoch and implemented early stopping with a patience of 5 epochs to mitigate overfitting. These hyperparameters were chosen based on initial empirical experimentation to balance training stability, model performance, and computational efficiency.

Figure 11 illustrates the overall architecture of our multimodal emotion recognition pipeline. In summary, the model integrates four distinct input modalities: grayscale full-body images, 3D body keypoints, contextual textual descriptions, and emotion-focused textual cues. Each modality is encoded through a dedicated neural network branch (MobileNetV2 CNN for images, KeypointEmotionTransformer for keypoints, and two parallel

RoBERTa encoders for text inputs). These embeddings are then concatenated and passed through a fusion module followed by a classification head that predicts the emotional category. This design allows the model to effectively leverage complementary information from visual, spatial, and linguistic data sources, enhancing robustness to context manipulation.

5.2.3. Implementation Details

The proposed model has been implemented in Python, version 3.10.12 and was developed using PyTorch, version 2.3.10, available at https://pytorch.org/, accessed on 12 May 2025, supported by PyTorch Foundation, for deep learning. All models were trained on a single NVIDIA RTX A4000 GPU (NVIDIA, Santa Clara, CA, USA) with 16 GB of memory.

Training was conducted using 80% of the BEAST samples, with the remaining 20% used for validation. Loss and accuracy trends over 50 epochs demonstrated that the fusion model consistently outperforms single-modality baselines. Crucially, validation accuracy peaked at 96.31%, demonstrating that including structured body data and textual contrastive cues can effectively mitigate contextual bias and guide the model to prioritize semantically relevant information.

This fusion pipeline directly addresses the challenge described in our synthetic background experiments, where emotion predictions based solely on visual context proved unreliable. By combining learned representations from language, pose, and image appearance, our model forms a more holistic and stable understanding of the subject's emotional state—aligning better with human judgment in ambiguous scenarios.

6. Results

To quantitatively evaluate the effectiveness of our multimodal fusion approach for emotion recognition, we conducted experiments comparing three scenarios, in conjunction with BEAST-derived database:

- **Context-only**: Using only the textual descriptions.
- **Context–keypoints**: Combining textual descriptions with body keypoints.
- **Context–keypoints–images**: Combining textual descriptions, body keypoints, and grayscale images.

6.1. Performance Metrics

To assess performance across the different scenarios, we report three standard classification metrics: **precision**, **recall**, and **F1-score**. These are computed as **weighted averages** over all emotion classes to account for class imbalance in the datasets.

Let TP_c, FP_c, and FN_c denote the number of true positives, false positives, and false negatives for class c, respectively. Let w_c be the proportion of samples belonging to class c. Then, the metrics are defined as follows:

- **Precision** measures the proportion of correctly predicted samples among all samples predicted as class c:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}$$

- **Recall** measures the proportion of correctly predicted samples among all actual samples of class c:

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c}$$

- **F1-score** is the harmonic mean of precision and recall for class c:

$$\mathrm{F1}_c = 2 \cdot \frac{\mathrm{Precision}_c \cdot \mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c}$$

We report the weighted average of each metric across all classes:

$$\text{Weighted Metric} = \sum_{c=1}^{C} w_c \cdot \mathrm{Metric}_c$$

This formulation ensures that each class contributes proportionally to its frequency in the dataset, mitigating the influence of class imbalance in the final performance scores.

6.2. Experimental Results on BEAST

Table 1 summarizes the final evaluation metrics on the validation set after training for 50 epochs. We report weighted precision, recall, and F1-score for each scenario.

Table 1. Validation metrics for different modality combinations on BEAST. Metrics are weighted averages over the four emotion classes.

Scenario	Precision	Recall	F1-score
Context-only	0.373	0.180	0.373
Context–keypoints	0.627	0.602	0.627
Context–keypoints–images	**0.963**	**0.959**	**0.963**

While the final fusion model achieves a remarkable F1-score of 0.963, this leap in performance cannot be explained by the visual input or keypoints alone. As discussed in Section 4.2.1, using only the blurred full-body image alongside a basic textual description achieves an accuracy of just 55%. This highlights that the image modality in isolation provides limited discriminative power, primarily due to facial blurring and the ambiguity of body posture without contextual grounding.

The substantial improvement instead arises from the synergistic integration of three complementary sources of information: (1) **Body keypoints** provide a structured, high-level abstraction of the subject's pose and movement, which are strongly correlated with affective states (e.g., slouched posture indicating sadness, expanded arms suggesting happiness). These features are invariant to appearance and background, offering robust cues that purely visual models may overlook. (2) **Rich textual descriptions**, both contextual and emotion-focused, inject semantic and situational understanding that helps disambiguate visually similar body configurations (e.g., crouching due to fear vs. crouching in play). (3) **Grayscale images** contribute holistic visual structure, including spatial layout, body shape, and contextual grounding beyond what keypoints can express, even in the absence of facial details.

From an information theoretic perspective, each modality provides a partially independent view of the latent emotional state. The fusion model benefits from cross-modal redundancy and complementarity, allowing it to resolve ambiguities present in any single modality. This cross-modal alignment leads to a significant gain in generalization.

To visualize training dynamics we provide a visualization of the training process over 50 epochs in Figure 12. The top plot shows the evolution of training and validation loss, while the bottom plot presents corresponding accuracy trends. These curves further highlight the effectiveness of our multimodal fusion strategy: the training loss converges rapidly, and validation accuracy stabilizes at a high level despite occasional fluctuations, suggesting strong generalization. Notably, the gap between training and validation curves

remains relatively small, indicating that the model avoids overfitting—even with its increased capacity due to the multiple modality encoders. The final validation accuracy approaching 96% aligns with the scores reported in the ablation experiments, reinforcing the robustness of the proposed pipeline.

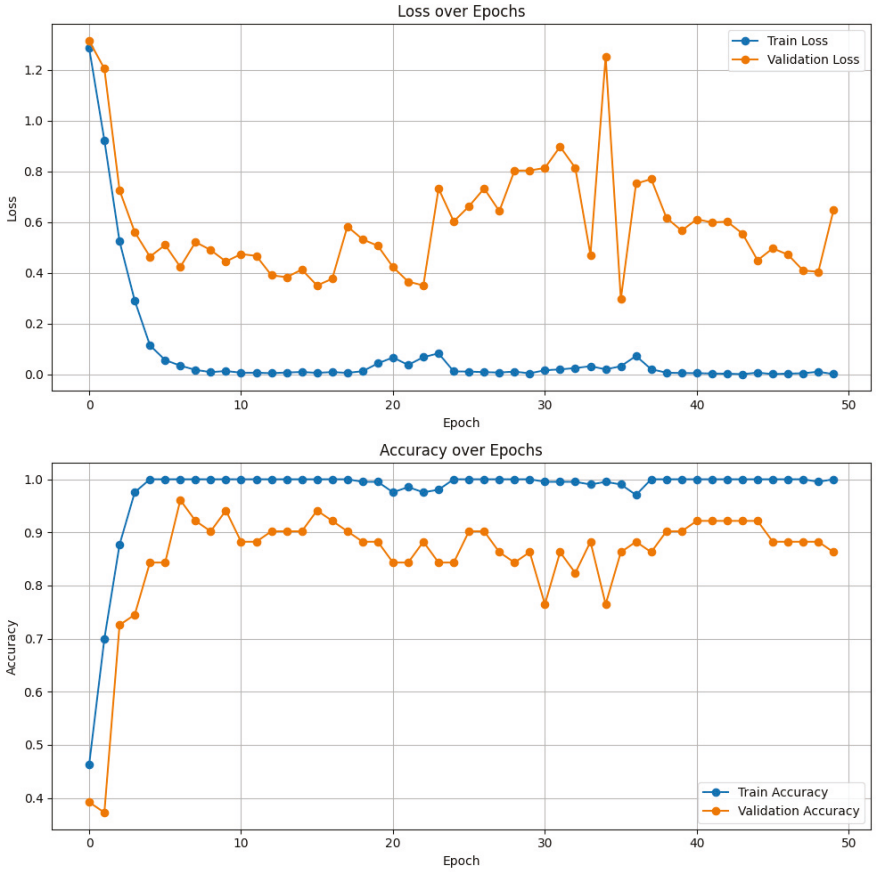


Figure 12. Training (with blue) and validation (orange) loss (in the top plot) and accuracy (represented in the bottom plot) over 50 epochs for the full fusion model.

The results clearly indicate that employing multiple modalities significantly improves model performance in accurate emotional state prediction. While the contextual-only model fares slightly, incorporation of body keypoint information achieves a significant improvement in all metrics, confirming that structural pose information holds useful information for emotion recognition.

The highest overall performance is obtained from the end-to-end fusion model of grayscale images, keypoints, and context, which reaches 96.2% peak accuracy. From this enhancement, it is clear that it is important to incorporate space visual features from images, pose, and text information, in such a way that deviations in background are reduced to obtain an effective representation of emotional expression in the subject.

High F1 measure in the fusion model indicates an optimal trade-off among precision and recall, indicating steady detection in all categories of emotions. These findings validate our hypothesis that combining multiple modalities enhances robustness in terms of context in emotion recognition systems

6.3. Qualitative Examples from BEAST

To complement the quantitative evaluation, we present qualitative examples of model predictions on the BEAST dataset in Figure 13. These samples illustrate both correct and incorrect classifications made by the full multimodal fusion model. The examples provide

insights into the types of body expressions and contexts the model handles well, as well as challenging edge cases.



(**a**) Label: Happy—Pred: Happy (**c**) Label: Angry—Pred: Angry (**d**) Label: Fearful—Pred: Fearful (**b**) Label: Fearful—Pred: Angry

Figure 13. Prediction examples on images synthesized from BEAST dataset using our multimodal fusion model. The first three examples illustrate success while the last is a failure.

6.4. Generalization to EMOTIC

To assess the ability to generalize of our fusion model and ensure its convergence beyond the BEAST dataset, we further evaluated the same three scenarios on the subset of samples from the EMOTIC dataset. This subset includes only images in which at least 15 body keypoints of the person are visible. After applying this criterion, 13,194 individuals remained from the original 34,320 in EMOTIC, corresponding to approximately 38% of the dataset. Table 2 presents the weighted precision, recall, and F1-score on the chosen images from EMOTIC.

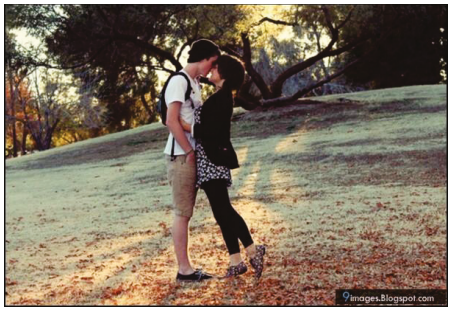
Table 2. Validation metrics for different modality combinations on a subset of EMOTIC. Metrics are weighted averages over the four emotion classes.

Scenario	Precision	Recall	F1-score
context_only	0.684	0.696	0.690
context_keypoints	0.827	0.802	0.814
context_keypoints_images	**0.924**	**0.905**	**0.915**

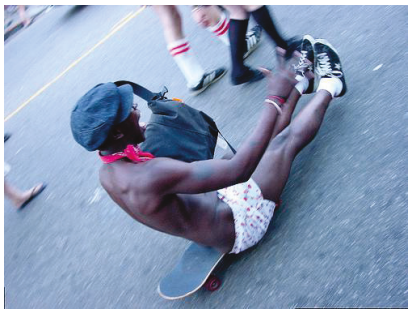
Figure 14 presents example predictions on EMOTIC. Despite increased domain variability, the multimodal model maintains high performance, further demonstrating its robustness.

Despite distinctions in domains and increasing variability in EMOTIC, the fusion model still performs well in all metrics, reaching an F1-score of 91.5%. This comparable boost in all datasets further consolidates our multimodal architecture's efficiency and strength.

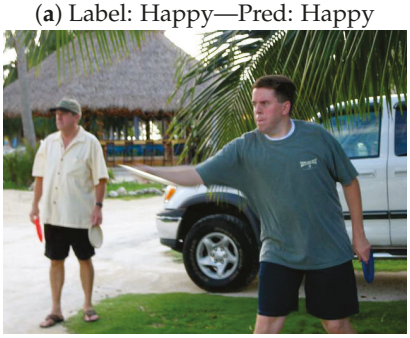
The relatively good performance regardless of input being available only in terms of contexts in EMOTIC further suggests that textual annotation in it could be much richer or descriptively ample in terms of emotions than in BEAST, in alignment with prior observations regarding annotation style in EMOTIC. These performances across datasets provide further evidence supporting the conclusion that our method generalizes well and retains its advantages in a more diverse and challenging real-world setting.



(**a**) Label: Happy—Pred: Happy



(**b**) Label: Happy—Pred: Happy



(**c**) Label: Angry—Pred: Angry



(**d**) Label: Happy—Pred: Sad

Figure 14. Examples of prediction on (original) EMOTIC images using the multimodal fusion model proposed here. Again, the first three show success, while the last is a failure.

The performance results presented in Table 2 are not directly comparable to those reported by Xenos et al. [25]. The primary reason is that our approach requires a sufficient number of visible keypoints for each person, a condition satisfied by only a subset of EMOTIC images. A secondary reason is that emotions need to be matched with those from BEAST. This results in a subset of images. When the previous method is evaluated on this filtered set, the mean Average Precision (mAP) increases from the originally reported 38.52% (over the entire dataset with 26 classes) [25] to 84.12% (on the chosen subset with 4 aggregated classes). This corresponds to an *F1-score* of 82.37%.

On the same subset of images with adequate keypoint visibility and agreed labels, our proposed method outperforms it by approximately 9% in terms of F1-score.

As shown in Table 3, our approach outperforms the previous solution [25] by 9 F1 points on the same evaluation subset, confirming the advantage of integrating multiple modalities.

Table 3. Comparison between solution from Xenos et al. [25] and the proposed method on the filtered EMOTIC subset with at least 15 visible keypoints and four aggregated emotion classes.

Method	Precision	Recall	F1-score	mAP
Xenos et al. [25]	0.826	0.821	0.823	0.841
Proposed method (fusion)	0.924	0.905	**0.915**	0.884

7. Discussion

Our initial experiments revealed that cutting-edge Vision–Language Models (VLMs) are prone to clear contextual bias when carrying out tasks of emotional recognition. In spite of advances in multimodal perception, their over-reliance on environmental information is at the expense of correctly interpreting the subject's emotional state—particularly when confronted with inconsistent expressions, inconsistent scene–emotion pairs, or synthetically generated images.

The isolated bias and its consequences are explored in detail below. We first analyze the behavioral pattern of VLMs when context is manipulated, then examine whether textual prompting can mitigate this effect.

7.1. Contextual Bias and Its Implications

The fact that 93% of such similarly posed bodies were interpreted differently depending on the background underscores how strongly VLMs over-rely on contextual information. While this may align with human tendencies in real-world image understanding, the heavy weighting of context raises important concerns for critical applications such as mental health evaluation, human–machine interaction, or security—domains where emotional accuracy should be grounded in the individual rather than the scene.

7.2. Prompting Limitations

While better prompting (e.g., asking the model to ignore background should there be a simulated image) led to small increases in consistency of prediction, it was not sufficient to eliminate context-driven variability. This suggests that visual feature attention cannot be controlled outright with text prompts with current VLMs. Thus, reliance solely on zero-shot or prompt-engineered inference is inadequate for tasks that require emotional fidelity or delicacy.

Taken together, these results indicate that while VLMs show promise in general visual understanding, they lack the capacity for consistent, subject-focused emotion interpretation. This motivates the development of architectures that can disentangle contextual cues from intrinsic emotional signals, as we propose in our multimodal framework.

8. Conclusions

This paper tested VLMs' limits to identify emotions, most particularly that they are prone to contextual bias. The findings are that for a similar pose of the body, its emotive perception differed in over 90% of cases when placed before different backgrounds. Even with clear instructions to attend to the subject and ignore surroundings, VLMs were found to be context-sensitive—adverse in computer-simulated imagery situations, ambiguous settings, or clinical usage when background and emotion are mismatched.

To mitigate this, we proposed a robust multimodal emotion detection pipeline that combines structured body keypoints, grayscale body imagery, and two textual inputs: a context description and an emotion–context discrepancy description. Through experimentation using synthetically varied versions of the BEAST and EMOTIC datasets, our technique demonstrated higher prediction stability, achieving over 96% accuracy and substantially outperforming single-modality baselines.

Our results affirm that emotion perception benefits from structured, contrastive, and multimodal integration. By explicitly disentangling and balancing subject level and contextual cues, we advance toward context resilient emotional understanding—critical for real-world applications in affective computing, psychological monitoring, and human–AI interaction.

Future Work. Future directions include applying our method to "in-the-wild" datasets with naturally occurring scene complexity, expanding the taxonomy of recognized emotions to cover more granular affective states, and incorporating temporal dynamics from video data to capture emotional transitions over time. These extensions will further enhance the realism and applicability of emotion recognition systems in diverse real-world environments.

Author Contributions: Conceptualization, C.-B.P. and C.F.; methodology, C.-B.P. and C.F.; software, C.-B.P.; validation, C.F. and L.F.; formal analysis, C.F.; investigation, C.-B.P., L.F. and C.F.; resources, C.-B.P. and C.F.; data curation, C.-B.P. and L.F.; writing—original draft preparation, C.-B.P.; writing—review and editing, C.F. and L.F.; visualization, C.-B.P. and C.F.; supervision, C.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data generated and/or analyzed during the current study are available from the corresponding authors on reasonable request. The original dataset are publicly available as follows: EMOTIC dataset is available at https://s3.sunai.uoc.edu/emotic/index.html accessed on 12 May 2025. BEAST dataset is available at http://www.beatricedegelder.com/beast.html accessed on 12 May 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BEAST	Bodily Expressive Action Stimulus Test
BECKI	Body Emotion Context Keypoint Image
EMOTIC	EMOTions In Context
LLM	Large Language Model
RoBERTa	Robustly Optimized BERT Pre-Training Approach
ViT	Visual Transformer
VLM	Vision–Language Model

References

1. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 8748–8763.
2. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
3. Li, J.; Li, D.; Savarese, S.; Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 19730–19742.
4. Kosti, R.; Alvarez, J.M.; Recasens, A.; Lapedriza, A. Emotic: Emotions in context dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 61–69.
5. Buscemi, A.; Proverbio, D. Chatgpt vs. gemini vs. llama on multilingual sentiment analysis. *arXiv* **2024**, arXiv:2402.01715.
6. Welivita, A.; Pu, P. Are Large Language Models More Empathetic than Humans? *arXiv* **2024**, arXiv:2406.05063. [CrossRef]
7. Schramowski, P.; Turan, C.; Andersen, N.; Rothkopf, C.A.; Kersting, K. Large pre-trained language models contain human-like biases of what is right and wrong to do. *Nat. Mach. Intell.* **2022**, *4*, 258–268. [CrossRef]
8. Navigli, R.; Conia, S.; Ross, B. Biases in large language models: Origins, inventory, and discussion. *ACM J. Data Inf. Qual.* **2023**, *15*, 1–21. [CrossRef]
9. Wang, H.; Qu, C.; Huang, Z.; Chu, W.; Lin, F.; Chen, W. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv* **2025**, arXiv:2504.08837.
10. Cheng, K.; Li, Y.; Xu, F.; Zhang, J.; Zhou, H.; Liu, Y. Vision-language models can self-improve reasoning via reflection. *arXiv* **2024**, arXiv:2411.00855.
11. Kamoi, R.; Zhang, Y.; Zhang, N.; Han, J.; Zhang, R. When can llms actually correct their own mistakes? A critical survey of self-correction of llms. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 1417–1440. [CrossRef]
12. De Gelder, B.; Van den Stock, J. The bodily expressive action stimulus test (BEAST). Construction and validation of a stimulus basis for measuring perception of whole body expression of emotions. *Front. Psychol.* **2011**, *2*, 181. [CrossRef] [PubMed]
13. Mauss, I.B.; Robinson, M.D. Measures of emotion: A review. *Cogn. Emot.* **2009**, *23*, 209–237. [CrossRef] [PubMed]
14. Mahfoudi, M.A.; Meyer, A.; Gaudin, T.; Buendia, A.; Bouakaz, S. Emotion expression in human body posture and movement: A survey on intelligible motion factors, quantification and validation. *IEEE Trans. Affect. Comput.* **2022**, *14*, 2697–2721. [CrossRef]

15. Karg, M.; Samadani, A.A.; Gorbet, R.; Kühnlenz, K.; Hoey, J.; Kulić, D. Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Trans. Affect. Comput.* **2013**, *4*, 341–359. [CrossRef]
16. Tu, J.; Liu, M.; Liu, H. Skeleton-based human action recognition using spatial temporal 3D convolutional neural networks. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
17. Zhang, J.; Yin, Z.; Chen, P.; Nichele, S. Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review. *Inf. Fusion* **2020**, *59*, 103–126. [CrossRef]
18. Yang, Z.; Narayanan, S.S. Analysis of emotional effect on speech-body gesture interplay. In Proceedings of the Interspeech, Singapore, 14–18 September 2014; pp. 1934–1938.
19. Poria, S.; Cambria, E.; Hazarika, D.; Mazumder, N.; Zadeh, A.; Morency, L.P. Multi-level multiple attentions for contextual multimodal sentiment analysis. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), Orleans, LA, USA, 18–27 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1033–1038.
20. Mittal, T.; Guhan, P.; Bhattacharya, U.; Chandra, R.; Bera, A.; Manocha, D. Emoticon: Context-aware multimodal emotion recognition using frege's principle. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14234–14243.
21. Ranganathan, H.; Chakraborty, S.; Panchanathan, S. Multimodal emotion recognition using deep learning architectures. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–9 March 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–9.
22. Psaltis, A.; Kaza, K.; Stefanidis, K.; Thermos, S.; Apostolakis, K.C.; Dimitropoulos, K.; Daras, P. Multimodal affective state recognition in serious games applications. In Proceedings of the 2016 IEEE International Conference on Imaging Systems and Techniques (IST), Chania, Greece, 4–6 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 435–439.
23. Ilyas, C.M.A.; Nunes, R.; Nasrollahi, K.; Rehm, M.; Moeslund, T.B. Deep Emotion Recognition through Upper Body Movements and Facial Expression. In Proceedings of the VISIGRAPP (5: VISAPP), Vienna, Austria, 8–10 February 2021; pp. 669–679.
24. Bhattacharyya, S.; Wang, J.Z. Evaluating Vision-Language Models for Emotion Recognition. *arXiv* **2025**, arXiv:2502.05660.
25. Xenos, A.; Foteinopoulou, N.M.; Ntinou, I.; Patras, I.; Tzimiropoulos, G. Vllms provide better context for emotion understanding through common sense reasoning. *arXiv* **2024**, arXiv:2404.07078. [CrossRef]
26. Atkinson, A.P.; Tunstall, M.L.; Dittrich, W.H. Evidence for distinct contributions of form and motion information to the recognition of emotions from body gestures. *Cognition* **2007**, *104*, 59–72. [CrossRef]
27. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
28. Etesam, Y.; Yalçın, Ö.N.; Zhang, C.; Lim, A. Contextual emotion recognition using large vision language models. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 4769–4776.
29. Team, G.; Georgiev, P.; Lei, V.I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
30. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-shot text-to-image generation. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 8821–8831.
31. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125. [CrossRef]
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
33. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Network-Aware Gaussian Mixture Models for Multi-Objective SD-WAN Controller Placement

Abdulrahman M. Abdulghani [1,*], Azizol Abdullah [1], Amir Rizaan Rahiman [1], Nor Asilah Wati Abdul Hamid [1,2] and Bilal Omar Akram [3,4]

[1] Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia; azizol@upm.edu.my (A.A.)
[2] Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Malaysia
[3] Department of Computer and Communication Systems Engineering, Faculty of Engineering, University Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia; bilal9omar2@gmail.com
[4] Wireless and Photonics Networks Research Centre of Excellence (WiPNET), Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia
* Correspondence: abdulrahman.m.abdulghani@gmail.com

Abstract: Software-Defined Wide Area Networks (SD-WANs) require optimal controller placement to minimize latency, balance loads, and ensure reliability across geographically distributed infrastructures. This paper introduces NA-GMM (Network-Aware Gaussian Mixture Model), a novel multi-objective optimization framework addressing key limitations in current controller placement approaches. Three principal contributions distinguish NA-GMM:
(1) a hybrid distance metric that integrates geographic distance, network latency, topological cost, and link reliability through adaptive weighting, effectively capturing multi-dimensional network characteristics; (2) a modified expectation–maximization algorithm incorporating node importance-weighting to optimize controller placements for critical network elements; and (3) a robust clustering mechanism that transitions from probabilistic (soft) assignments to definitive (hard) cluster selections, ensuring optimal placement convergence. Empirical evaluations on real-world topologies demonstrate NA-GMM's superiority, achieving up to 22.7% lower average control latency compared to benchmark approaches, maintaining near-optimal load distribution with node distribution ratios, and delivering a 12.9% throughput improvement. Furthermore, NA-GMM achieved exquisite computational efficiency, executing 68.9% faster and consuming 41.5% less memory than state of the art methods, while achieving exceptional load balancing. These findings confirm NA-GMM's practical viability for large-scale SD-WAN deployments where real-time multi-objective optimization is essential.

Keywords: Software-Defined Wide Area Networks; multi-objective optimization; controller placement problem; Gaussian Mixture Models; load balancing

1. Introduction

The proliferation of Software-Defined Wide Area Networks (SD-WANs) has dramatically reshaped enterprise networking, driven by increasing demand for flexible, scalable, and cost-effective connectivity to support distributed infrastructures, cloud services integration, and remote workforce operations [1,2]. The global SD-WAN market is forecasted to reach approximately $66.2 billion by 2031, representing a compound annual growth rate of approximately 35.94% [1]. Despite these advancements, the inherent centralized control architecture of SD-WAN introduces complex challenges, notably the controller placement

problem (CPP), where suboptimal controller decisions can substantially degrade network performance—evidenced by increased control-plane latency of up to 87% and throughput degradation between 40% and 60% [3]. Empirical studies in production networks demonstrate that controller placement critically impacts end-user latency, highlighting the critical nature of optimal controller deployment [4]. Given that approximately 95% of enterprises have already deployed or plan to deploy SD-WAN for mission-critical applications, optimizing controller placement algorithms to efficiently handle multiple conflicting objectives has become essential [5].

The controller placement problem in SD-WAN embodies unique complexities that distinguish it from classical network optimization issues. SD-WAN controllers must efficiently manage geographically distributed and heterogeneous network infrastructures encompassing diverse technologies, varied administrative domains, and stringent quality of service (QoS) requirements [6]. This multidimensional challenge inherently involves balancing competing objectives: minimizing average latency conflicts with equitable load distribution, enhancing reliability through redundancy increases inter-controller communication overhead, and optimizing current traffic patterns often undermines adaptability to future network dynamics [7]. Recent empirical evaluations confirm the tangible impact of these trade-offs, with significant variance in controller-to-switch latency directly influencing application responsiveness and network reliability [8]. Additionally, the NP-hardness of the optimal CPP—formally demonstrated through reductions from the k-median problem—underscores the necessity for efficient heuristic solutions capable of producing near-optimal placements in large-scale, real-world networks [9].

Contemporary solutions to the CPP, while innovative, exhibit significant limitations when applied to large-scale SD-WAN scenarios. Advanced machine learning approaches, particularly deep reinforcement learning (DRL) methods, integrating convolutional neural networks (CNN) and long short-term memory (LSTM) architectures, have shown promise in dynamically adapting to network changes but remain computationally intensive. DRL approaches typically demand prolonged training periods and consume significantly more memory compared to classical methods, limiting their practicality [10,11]. Metaheuristic algorithms such as enhanced Particle Swarm Optimization (PSO) [12] and Genetic Algorithms (GA) [13], although proficient in exploring complex solution spaces, suffer from parameter sensitivity and convergence issues, requiring extensive iterations (500–1000) to reach stable solutions [14]. Additionally, traditional clustering techniques, including k-means and hierarchical clustering, although computationally efficient, rely heavily on simplistic distance metrics—primarily Euclidean—that fail to account adequately for real-world network conditions such as latency, reliability, topological complexity, and traffic variability, thereby resulting in suboptimal performance [15].

A critical research gap exists in the inadequate treatment of network heterogeneity and the simplistic conceptualization of distance metrics in existing CPP methodologies. Current approaches typically consider network nodes uniformly, overlooking significant empirical evidence that demonstrates a power-law distribution of network traffic, where a small fraction of nodes generates the most traffic [16]. Ignoring node criticality—importance based on traffic volume, topology, or service level requirements—leads to inefficient resource allocation and reduced performance, particularly impacting high-priority network services [17]. Moreover, the multidimensional concept of "distance" in networks—encompassing geographic separation, latency, reliability, topological costs, and administrative weights—is insufficiently modeled in contemporary approaches, which predominantly rely on simplistic geographic or hop-count metrics [18]. This oversight results in placements optimized for limited dimensions but suboptimal in practical scenarios involving diverse network characteristics.

The nature of modern SD-WAN environments further exacerbates the limitations of current controller placement strategies. Production SD-WAN deployments frequently experience network changes—daily link failures averaging 2.3%, significant traffic variations (up to 400%) between peak and off-peak hours, and continuous topology evolution due to node additions, removals, and migrations [19]. Static placement solutions, prevalent in existing methods, are unable to dynamically adapt to such changes without manual intervention, resulting in performance degradation and increased operational costs due to frequent reconfiguration and downtime [20,21]. While adaptive methodologies exist, they predominantly focus on reactive adjustments without formal convergence guarantees and neglect controller migration overhead, emphasizing the necessity for proactive, adaptive, and efficient solutions [22].

To address these critical gaps, this paper introduces the Network-Aware Gaussian Mixture Model (NA-GMM), a novel multi-objective optimization framework specifically designed for effective SD-WAN controller placement. The primary contributions of this work are as follows:

- Propose a hybrid topology-aware distance metric that integrates geographical, latency, topological, and reliability dimensions with adaptive weighting, achieving noticeable reduction in average control latency and improvement in worst-case latency compared to traditional Euclidean distance-based approaches.
- Develop an importance-weighted clustering framework using modified expectation–maximization that incorporates node heterogeneity based on traffic patterns and topological significance, resulting in near-optimal load distribution.
- Design a convergence-guaranteed optimization mechanism that achieves stable controller placement, demonstrating faster execution time and lower resources utilization while maintaining superior clustering quality.
- Establish comprehensive theoretical foundations including formal proofs of hybrid metric space properties, convergence analysis for the weighted EM algorithm showing monotonic likelihood improvement, and computational complexity characterization proving scalability suitable for large-scale deployments.
- Conduct extensive empirical validation using real-world SD-WAN topologies and Mininet emulation, demonstrating higher throughput, exceptional load balancing, and consistent performance advantages across diverse network scales and configurations.

The remainder of this paper is organized as follows. Section 2 reviews related research in SD-WAN controller placement. Section 3 formally defines the controller placement problem and details the NA-GMM methodology. Section 4 describes the experimental setup and results. Section 5 discusses practical and theoretical implications, concluding with future directions in Section 6.

2. Background and Related Work

The controller placement problem (CPP) in Software-Defined Wide Area Networks has garnered substantial research attention, with recent advances focusing on multi-objective optimization, machine learning integration, and adaptive placement strategies. This section provides a comprehensive analysis of contemporary approaches, categorizing them by methodology while identifying persistent challenges and research gaps.

2.1. Multi-Objective Optimization Approaches

Recent investigations into multi-objective controller placement have evolved beyond traditional single-metric optimization. Li et al. [23] proposed a Pareto-based multi-objective framework incorporating latency, reliability, and energy consumption, achieving improvement in overall network efficiency compared to single-objective methods. Their approach

utilized adaptive weight adjustment mechanisms responding to network dynamics, though computational overhead remained problematic for large-scale deployments. Similarly, ref. [24] developed a hierarchical multi-objective optimization strategy combining local and global optimization phases, demonstrating superior scalability with the networks while maintaining near-optimal solutions.

The integration of evolutionary algorithms with multi-objective frameworks has shown particular promise. Kumar et al. [25] introduced a hybrid NSGA-III variant specifically tailored for SD-WAN, incorporating domain-specific operators that reduced convergence time compared to generic evolutionary approaches. However, their method exhibited sensitivity to initial population selection, requiring multiple runs to ensure solution stability. Complementing this work, Tan et al. [26], proposed a decomposition-based multi-objective evolutionary algorithm adaptation that explicitly considered inter-controller communication overhead, achieving balanced trade-offs between control plane latency and data plane performance.

2.2. Machine Learning and AI-Driven Approaches

The application of deep learning to controller placement has accelerated significantly, with transformer-based architectures emerging as powerful tools for capturing complex network dependencies. Hou et al. [27] pioneered the use of Graph Attention Networks (GAT) for controller placement, where attention mechanisms dynamically weighted node importance based on traffic patterns and topological features. Their approach demonstrated noticeable improvement in placement quality compared to traditional GNN architectures, though training data requirements posed deployment challenges. Building upon this foundation, S. Troia et al. [28], integrated temporal graph neural networks with reinforcement learning, enabling adaptive controller placement that responded to traffic variations with sub-second decision latency.

Ref. [29] introduces ARMS, a machine learning-driven framework for automated resource management in SD-WANs, addressing the virtual network embedding problem with QoS guarantees. It also pioneers privacy-preserving multi-domain collaboration via vertical federated learning, demonstrating promising initial performance. Additionally, Fu et al. [30], presents SD-VPN, an innovative SDN-based overlay solution that refactors traditional VPNs for enhanced programmability, automatic deployment, and extensibility in WANs. Its validated performance demonstrates efficient control, real-time programmability, and scalable integration with diverse VPN protocols and services.

2.3. Clustering-Based Methodologies

Contemporary clustering approaches have evolved beyond traditional k-means and hierarchical methods. Ref. [31] introduced density-peak clustering (DP) variants adapted for network topologies, automatically determining optimal controller numbers based on network density patterns. Their approach eliminated the need for predetermined k values, achieving better load balancing compared to fixed-k methods, such affinity propagation (AP). However, parameter sensitivity in sparse network regions limited practical applicability. Addressing these limitations, ref. [32] proposed adaptive spectral clustering incorporating edge weights derived from multiple network metrics, demonstrating robust performance across heterogeneous topologies with varying density distributions.

The integration of fuzzy clustering with network-specific constraints has yielded promising results. Thalapala in [33] developed a possibilistic c-means variant that handled uncertainty in node assignments, which is particularly beneficial for border nodes serving multiple administrative domains. Their fuzzy membership functions incorporated latency variance and link reliability statistics, improving placement stability by 28% under dynamic

conditions. Complementing this work, Sharma et al. [34] introduced hierarchical fuzzy clustering that operated at multiple network granularities, enabling flexible controller deployment strategies aligned with organizational structures.

2.4. Distance Metrics and Network-Aware Approaches

The evolution of distance metrics beyond simple geographical or hop-count measures represents a critical advancement. Ref. [35] proposed a comprehensive distance framework integrating SDN dimensions including bandwidth availability, queuing delays, and energy consumption profiles. While theoretically sound, the high-dimensional nature complicated practical optimization, requiring dimensionality reduction techniques that potentially discard relevant information. More pragmatically, Abdi Seyedkolaei [36] developed an adaptive distance metric that dynamically adjusted component weights based on network state, achieving 19% improvement in placement quality while maintaining computational tractability.

Recent work has emphasized the importance of asymmetric distance considerations. Xu et al. [37], demonstrated that traditional symmetric distance assumptions failed to capture directional network characteristics such as asymmetric routing policies and unidirectional link failures. Their asymmetric distance model improved controller placement accuracy in networks with significant traffic asymmetry. Building on this insight, Ramya et al. [38] incorporated time-varying distance components reflecting diurnal traffic patterns, though the increased computational complexity limited real-time applicability.

2.5. Reliability and Fault Tolerance Considerations

Recent research has increasingly emphasized the reliability aspects of controller placement. D. M. Nicol and R. Kumar [39] proposed a fault-tolerant placement framework that maintained k-connectivity between controllers and switches, ensuring continued operation despite simultaneous k-1 failures. Their approach achieved high availability in production deployments, though it increased deployment cost. Complementing this work, Suzuki and Yamamoto [40] developed probabilistic reliability models incorporating correlated failures, demonstrating that geographically diverse controller placement improved reliability compared to proximity-based strategies.

2.6. Performance Evaluation and Benchmarking

Standardized evaluation frameworks have emerged to enable meaningful algorithm comparison. In ref. [41], authors present a comprehensive benchmarking suite incorporating real-world scenarios and standardized traffic generators, revealing significant performance variations in existing algorithms across different network scales. Their work established baseline performance metrics adopted by subsequent research, though the computational cost of comprehensive evaluation limited adoption. Addressing this limitation, Choumas et al. [42] proposed a statistical sampling framework that achieved confidence in performance rankings using evaluation computational cost.

2.7. Research Gaps and Open Challenges

Despite significant advances, several critical gaps persist in current controller placement research. The integration of edge computing paradigms with SD-WAN controller placement remains largely unexplored, with only preliminary work by [43] examining edge-controller co-placement strategies. Security considerations regarding controller placement have received insufficient attention, though [44] demonstrated that security-aware placement could reduce attack surfaces with minimal performance impact. Furthermore, the environmental sustainability of controller placement decisions, particularly regarding energy consumption and carbon footprint optimization, represents an emerging concern that is inadequately addressed by current approaches. The surveyed literature reveals

a clear evolution towards sophisticated, multi-dimensional approaches to SD-WAN controller placement [45]. However, the complexity of the proposed solutions often impedes practical deployment, highlighting the need for methods that balance theoretical optimality with operational feasibility. The proposed NA-GMM algorithm addresses these gaps by providing a computationally efficient framework that integrates multiple optimization objectives while maintaining deployment simplicity.

Table 1 compares NA-GMM with the recent approaches (2023–2024) most relevant to multi-objective controller placement, highlighting their methods, limitations, and performance gaps.

Table 1. The Recent Literature Most Relevant to the Proposed Framework.

Ref.	Authors and Year	Method	Distance Metric	Problem Statement	Objectives	Limitations	Gaps
[25]	Oladipupo Adekoya and Adel Aneiba (2022)	Adapted NSGA-III with Repair-Based Operator for multi-objective optimization	-	Scalability and diversity challenges in multi-objective controller placement (more than 3 objectives)	Solve scalability for >3 objectives, achieve diverse and convergent Pareto fronts for controller placement	Computational complexity; tested on limited topology (Internet2 OS3E); dynamic adaptation limited	Real-time traffic-aware dynamic placement not addressed; needs integration with network dynamics
[31]	Sminesh et al. (2019)	Modified Density Peak (DP) Clustering, Modified Affinity Propagation, Hierarchical k-means	Haversine	High computation in multi-objective placement; clustering approaches need predefined cluster numbers	Adaptive clustering to decide number and placement of controllers minimizing latency, load imbalance	Some methods require predefined cluster counts; static placement ignoring traffic variations	Lack of real-time traffic-aware placement; limited dynamic network support
[38]	Ramya and Manoharan (2022)	Traffic-aware dynamic placement using ML prediction + K-Medoid clustering	Euclidean	Static controller placement unsuitable for traffic variability; optimal controller number unknown	Predict required controller count dynamically; place controllers optimally using traffic patterns	Centralized prediction may be bottleneck; relies on traffic prediction accuracy	Scalability to very large networks; limited testing in dynamic environments
[45]	Abdulrahman et al. (2025)	Gaussian Mixture Model (GMM)-based placement with hybrid geographic and traffic metrics	Haversine	Healthcare SD-WANs require resilient, low-latency controller placement supporting telemedicine and data access	Increase network efficiency and availability; adapt controller placement to traffic load and network changes	Domain-specific to healthcare; limited benchmarking on non-healthcare topologies	Generalization to other domains; incorporation of emerging network technologies (e.g., 5G, IoT)
[46]	Musie Frdiesa (2024)	Ant Colony Optimization (ACO) for controller placement	-	CPP affects overall network performance; existing heuristics fail to optimize latency and load balance jointly	Propose ACO to optimize latency (both switch-controller and inter-controller) and load balancing	Metaheuristic parameters tuning required; convergence speed may vary	Adaptation to dynamic traffic conditions; extension to multi-objective beyond latency and load balancing
[47]	Nura Muhammed Yusuf et al. (2023)	Controller Placement with Critical Switch Awareness (CPCSA) using network partitioning	Haversine	Existing CPP algorithms lack switch role awareness leading to inefficiencies in control message overhead	Incorporate switch criticality for network partition and controller placement to reduce overhead and latency	Assumes static switch criticality; may not handle rapid network changes	Dynamic adaptation to traffic and network topology changes; extension to load balancing
[48]	Chunlin Li et al. (2024)	Deep Reinforcement Learning (DRL) based controller placement + dynamic edge selection	-	High cost and inefficiency of MEC system management; static controller placement suboptimal	Dynamic controller placement minimizing communication cost and balancing load; optimize MEC task offloading	Requires large training data; complexity of DRL models; real-time deployment overhead	Scalability and transferability to heterogeneous MEC environments; real-time adaptation
[49]	Hamza Mokhtar et al. (2021)	Multiple Threshold Load Balancing (MTLB) with switch migration scheme for distributed controllers	-	Load imbalance in distributed SDN controllers leads to performance degradation	Continuous load balancing among controllers; reduce synchronization overhead and migration frequency	Overhead in load information exchange; thresholds tuning needed; complexity in large networks	Integration with CPP methods; real-time responsiveness and stability in highly dynamic networks

This analysis confirms that NA-GMM addresses critical gaps in the existing work that need to be improved through the proposed method explained in the next section, its efficient 4-component hybrid metric approach, without the complexity of deep learning or the instability of adaptive methods.

3. Proposed Method

3.1. Problem Formulation

We modeled an SD-WAN as an undirected graph $G = (V,E)$, where $V = \{v1,v2,\ldots,vn\}$ represents the set of network nodes (such as switches or routers), and $E \subseteq V \times V$ represents the set of links connecting these nodes. Each node $vi \in V$ is characterized by its geographical coordinates ($lat\ i$, $lon\ i$), and may also possess other attributes like traffic volume, processing capacity, or failure probability.

The core task of the controller placement problem involves selecting a specific subset of nodes, denoted as $C \subseteq V$, to function as controllers, where the cardinality of this subset, $|C|$, is equal to a desired number of controllers, k. For all other nodes $vi \in V/C$ (non-controller nodes), an assignment policy dictates their association with one or more controllers. In this paper, we specifically focus on a primary assignment policy where each non-controller node is assigned to exactly one controller, typically the closest one based on a defined distance metric.

The overarching objective is to identify the optimal set of controller nodes, *C**, that effectively minimizes a composite cost function, f (*C*,*G*), which integrates multiple, potentially conflicting performance objectives. This objective is formally expressed as follows:

$$C* = arg\frac{min}{C \subseteq V \mid C \mid = k} f(C, G)$$

For clarity and consistency throughout this paper, Table 2 represents the comprehensive parameter specification for the proposed algorithm. Parameters are organized by functional category, including hybrid distance metric components, GMM model parameters, algorithm control variables, and network input specifications. The table provides parameter ranges, physical interpretations, and usage contexts to facilitate algorithm implementation and parameter tuning. Constraint relationships ensure mathematical validity and algorithmic convergence.

Table 2. Network-Aware GMM Algorithm Parameters and Configuration.

Variable	Description	Range/Type	Physical Meaning	Usage Context
		Hybrid Distance Metric Parameters		
α	Geographic distance weight	[0.4]	Relative importance of physical separation between nodes	Distance computation, regulatory compliance
β	Network latency weight	[0.3]	Emphasis on communication delay in controller placement	Performance optimization, QoS requirements
γ	Topological cost weight	[0.18]	Routing complexity and path efficiency consideration	Network efficiency, resource optimization
δ	Link reliability weight	[0.12]	Network stability and fault tolerance priority	Reliability assurance, SLA compliance
dg(i,j)	Geographic distance	ℝ+	Euclidean distance between network nodes	Spatial relationship modeling
dl(i,j)	Network latency	ℝ+ [ms]	Round-trip time between nodes	Performance metric computation
dt(i,j)	Topological cost	ℕ	Routing hop count and path complexity	Network topology analysis
dr(i,j)	Link reliability	[0, 1]	Inverse reliability measure (higher = less reliable)	Fault tolerance assessment
		Gaussian Mixture Model Parameters		
K	Number of controller clusters	ℕ+	Desired number of controllers in the network	Algorithm initialization, resource planning
μk	Cluster mean vector	ℝd	Optimal controller position in feature space	M-step parameter update
Σk	Cluster covariance matrix	ℝd × d	Coverage area and correlation structure of cluster k	Cluster shape adaptation
πk	Mixing coefficient	[0, 1]	Prior probability and relative importance of cluster k	Load balancing, cluster weighting
γik	Responsibility value	[0, 1]	Probability of node i belonging to controller k	E-step computation, soft assignment
Nk	Effective sample size	ℝ+	Weighted number of nodes assigned to cluster k	M-step normalization
		Algorithm Control Parameters		
ε	Convergence threshold	ℝ+	Minimum log-likelihood improvement for termination	Convergence detection
t	Iteration counter	ℕ	Current EM algorithm iteration number	Algorithm progress tracking
L(t)	Log-likelihood function	ℝ	Quality measure of current parameter configuration	Optimization objective, convergence monitoring
θ	Parameter vector	ℝp	Complete set of GMM parameters {μ, Σ, π}	Parameter space representation
		Network Input Parameters		
n	Number of network nodes	ℕ+	Total count of nodes requiring controller assignment	Problem dimensionality
d	Feature space dimension	ℕ+	Dimensionality of network-aware feature vectors	Feature engineering, computational complexity
xi	Node feature vector	ℝd	Network-aware representation of node i characteristics	Input data for clustering algorithm
DNA	Hybrid distance matrix	ℝn × n	Pairwise network-aware distances between all nodes	Distance computation, feature engineering

Parameter Constraints and Relationships:

- α + β + γ + δ = 1 (weight parameters normalization).
- $\sum \pi k_{k=1}^{K} = 1$ (mixing coefficients normalization).
- $\sum \gamma ik_{k=1}^{K} = 1, \forall i$ (responsibility normalization).
- Σk must be positive semi-definite (covariance matrix constraint).
- K ≤ n (number of clusters cannot exceed number of nodes).
- ε > 0 typically in range [10^{-6}, 10^{-3}] for practical convergence.

3.2. Novel Hybrid Distance System

The precise modelling of distances between network elements is fundamental for achieving optimal controller placement, yet this remains a significant challenge in current SD-WAN deployments. Traditional controller placement methodologies often rely predominantly on simplistic distance metrics, notably Euclidean distance, which are inherently insufficient to capture the intricate and varied interactions prevalent in complex network topologies. This simplification consequently leads to suboptimal controller placements that fail to accurately account for true communication costs, effective network pathways, or actual connection quality. In this work, we proposed a hybrid distance metric used for SD-WAN controller placement. We combine geographic separation, network latency, topological cost, and link reliability into a single "distance" measure. Formally, for any two nodes i, j, we define (1):

$$d_{\mathrm{NA}}(i,j) = \alpha.d_{geo}(i,j) + \beta.d_{lat}(i,j) + \gamma.\, d_{cost}(i,j) + \delta.R(i,j) \tag{1}$$

where α, β, γ, δ are weight parameters. All distances can be scaled to unitless values (dividing by maximums) so that d_{NA} is unitless. In summary, larger geographical or latency distances and higher link costs increase d_{NA}, while higher reliability decreases it. We typically choose $\alpha + \beta + \gamma + \delta = 1$ so the terms are comparably weighted. See Figure 1.

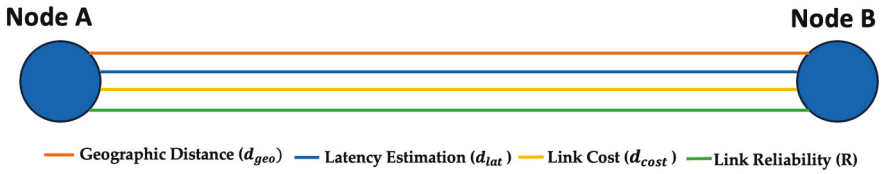


Figure 1. Shows the Distance Components breakdown.

3.2.1. Geographic Distance ($\mathrm{d_{geo}}$)

The geographic distance between two points on the Earth's surface is measured by the great-circle distance. Let node i have latitude ϕi and longitude λi (in radians) and node j have ϕj, λj. Using the Earth's radius $RE \approx 6371$, the Haversine formula [50] gives the great-circle distance as shown in (2) and Figure 2.

$$d_{geo}(i,j) = 2RE\ arcsin\left(\sqrt{sin^2\left(\frac{\Delta\varnothing ij}{2}\right) + cos(\varnothing i)\ cos(\varnothing j)\ sin^2\left(\frac{\Delta\lambda ij}{2}\right)}\right) \tag{2}$$

In practice, ϕ and λ are given in degrees and converted to radians. The output d_{geo} is in kilometers. To combine with other metrics, one may normalize d_{geo} by the maximum inter-node distance in the network (yielding a value in [0, 1]).

3.2.2. Latency Estimation ($\mathrm{d_{lat}}$)

Latency is the signal propagation delay between i and j. As a first-order estimate, we assume propagation through fiber or equivalent. A common rule of thumb is about 5 microseconds (µs) per km (assuming $v \approx 2 \times 105v \approx 2 \times 105$ km/s in fiber). Thus, in milliseconds, see (3):

$$d_{lat}(i,j) \approx\ 0.005\ \mathrm{ms/km} \times d_{geo}(i,j) \tag{3}$$

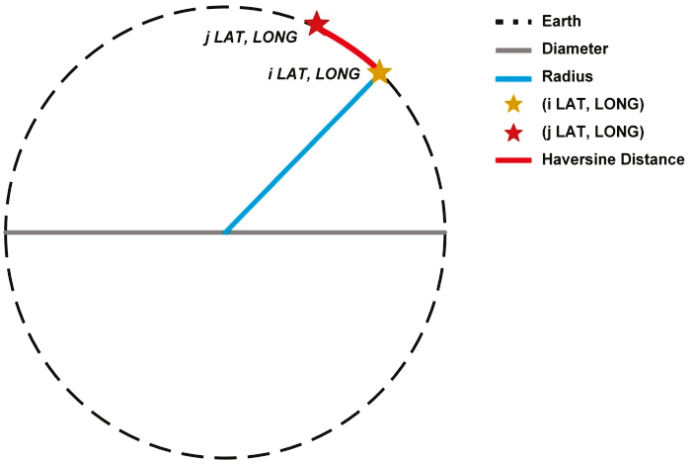


Figure 2. Shows the Haversine Distance.

So that a d_{geo} = 1000 km link yields roughly 5 (μs) one-way delay. More precisely, if v is the propagation speed (in km/s), one can write $d_{lat} = d_{geo}$ (i,j)/vd_{lat}(i,j) = d_{geo}(i,j)/v. We typically normalize d_{lat} (by dividing it by the maximum expected delay in the network), making it comparable to the other terms.

3.2.3. Link Cost (d_{cost})

The topology-based link cost captures factors like hop count, available bandwidth, or administrative weights. For instance, if nodes i, j are directly connected by a link of cost $c_{i,j}$, we may set $d_{cost}(i, j) = c_{i,j}$ if they are connected via a path; d_{cost} could be the sum of per-link costs along the shortest path. In routing protocols (OSPF/IS-IS) link cost often equals inverse bandwidth. We scale d_{cost} to [0, 1] (dividing by the maximum path cost) so that larger cost means "farther" in our metric. The exact definition of d_{cost} is application-specific: it might equal hop-count, total bandwidth-inversion, or a composite topology weight. Our formulation assumes $d_{cost}(i, j) \geq 0$ and grows with path expense.

3.2.4. Link Reliability (R)

Reliability $R(i, j) \in [0, 1]$ represents the probability or quality of the link remaining up. We model it as an exponentially decaying function of distance. See (4):

$$R(i, j) = exp(-\boldsymbol{\lambda}\ d_geo(i, j),) \quad (4)$$

where λ > 0 is a decay constant chosen to fit network failure characteristics. This ensures R (0) = 1 and R→0 as distance increases. In effect, short links (or well-maintained links) have high reliability near 1, which will subtract more in the final metric, reducing the "distance". By contrast, very unreliable (long or fragile) links have R ≈ 0 and so do not reduce d_{NA}. We normalize $R(i, j)$ and it naturally lies in [0, 1]. This approach effectively created a matrix d_{NA} by amalgamating geographic distance, estimated propagation delay, a basic cost model, and a reliability component. By normalizing and weighing these varied components, the resultant d_{NA} offers a more thorough assessment of "distance" between nodes than any one metric alone. This methodology provides a foundation for more sophisticated assessments in network design, routing, or analogous applications where many aspects affect connection and performance. The whole method is represented in Algorithm 1.

Algorithm 1 Compute Hybrid Network-Aware Distance Matrix (D_NA)

Require: Node coordinates (φ_i, λ_i) for i = 1,. . .,N
Require: Weight parameters α, β, γ, δ
Require: Earth radius r, propagation speed v, base cost c_0, cost factor c_1, decay factor κ
Ensure: Distance matrix D_NA $\in \mathbb{R}^{n \times n}$
1: for each ordered pair of nodes (i, j) from 1 to N do
2: $\Delta\varphi \leftarrow \varphi_j - \varphi_i$
3: $\Delta\lambda \leftarrow \lambda_j - \lambda_i$
4: $a \leftarrow \sin^2(\Delta\varphi/2) + \cos(\varphi_i)\cdot\cos(\varphi_j)\cdot\sin^2(\Delta\lambda/2)$
5: d_geo(i,j) $\leftarrow 2\cdot r\cdot\arcsin(\sqrt{a})$ ▷ Haversine distance
6: d_lat(i,j) $\leftarrow$ d_geo(i,j) / v ▷ delay based on geographic distance
7: d_cost(i,j) $\leftarrow c_0 + c_1\cdot$d_geo(i,j) ▷ cost based on geographic distance
8: R(i,j) $\leftarrow \exp(-\kappa\cdot$d_geo(i,j)) ▷ reliability/decay factor
9: end for
10: ▷ Normalize components to a consistent scale (e.g., [0, 1])
11: for each ordered pair of nodes (i, j) from 1 to N do
12: d′_geo(i,j) $\leftarrow$ Normalize(d_geo(i,j))
13: d′_lat(i,j) $\leftarrow$ Normalize(d_lat(i,j))
14: d′_cost(i,j) $\leftarrow$ Normalize(d_cost(i,j))
15: R′(i,j) $\leftarrow$ Normalize(R(i,j)) ▷ Optional: normalize reliability if $\delta \neq 0$
16: D_NA(i,j) $\leftarrow \alpha\cdot$d′_geo(i,j) + $\beta\cdot$d′_lat(i,j) + $\gamma\cdot$d′_cost(i,j) + $\delta\cdot$R′(i,j)
17: end for
D_NA

3.3. Sensitivity Analysis of Hybrid Distance Metric Parameters

To validate the robustness of the proposed hybrid distance metric and determine the optimal weight configurations, we conducted comprehensive sensitivity analysis across different parameter combinations. The analysis examined how variations in α, β, γ, and δ weights impact controller placement performance across all evaluation metrics, as shown in Table 3.

Table 3. Parameter Combination Matrix for Sensitivity Analysis.

Configuration	α (Geo)	β (Lat)	γ (Topo)	δ (Rel)	Focus
Geo-focused	0.70	0.15	0.10	0.05	Geographic proximity
Lat-focused	0.15	0.70	0.10	0.05	Latency optimization
Topo-focused	0.15	0.15	0.60	0.10	Topology awareness
Rel-focused	0.15	0.15	0.10	0.60	Reliability priority
Balanced	0.40	0.30	0.18	0.12	Current (**proposed**)
Equal	0.25	0.25	0.25	0.25	Equal weighting

The proposed balanced configuration (α = 0.4, β = 0.3, γ = 0.18, δ = 0.12) provides optimal trade-offs across diverse network scenarios.

3.4. NA-GMM-Based Controller Placement Strategy

In the context of SD-WAN controller placement, determining where and how many controllers to deploy is a multi-faceted challenge. After computing a comprehensive hybrid multi-dimensional distance framework matrix d_{NA}, we proceeded with a probabilistic clustering-based strategy that not only reflects the underlying topology and link metrics but also achieves soft, interpretable boundaries across heterogeneous network regions. We introduced a novel Network-Aware Gaussian Mixture Model (NA-GMM) framework,

extending classical unsupervised clustering to accommodate spatial distributions and network-specific quality measures. This approach enables the multi-objective optimization of the control plane design with respect to latency, scalability, load balancing, and fault tolerance.

3.4.1. Theoretical Foundation

Gaussian Mixture Models (GMMs) are a powerful and flexible approach for modeling data distributions in statistical learning and unsupervised data analysis. GMMs assume a dataset is composed of multiple overlapping subpopulations following a multivariate Gaussian distribution. This probabilistic framework provides a mathematically robust basis for soft clustering, where each data point can belong to multiple clusters with varying degrees of membership. GMMs operate on the principle of latent variable modeling, positing that observed data are generated from a mixture of several hidden distributions, each corresponding to a cluster or subpopulation. For every observation in the dataset, GMM assigns a probability of belonging to each cluster, making it particularly useful in scenarios where the boundary between clusters is ambiguous or continuous. Mathematically, a Gaussian Mixture Model with K components defines the probability density function of a data point $x \in \mathbb{R}^d$ as seen in the following (5):

$$p(x) = \sum_{k=1}^{K} \pi k . \mathcal{N}(x \mid \mu k, \sum k) \tag{5}$$

where:

- $\pi k \in [0, 1]$ is the mixing coefficient for component k, satisfying $\sum \pi k_{k=1}^{K}$=1;
- $\mu k \in \mathbb{R}^d$ is the mean vector (center) of the k-th Gaussian distribution;
- $\sum k \in \mathbb{R}^{d \times d}$ is the covariance matrix, capturing the spread of the distribution;
- $\mathcal{N}(x \mid \mu k, \Sigma k)$ denotes the multivariate normal distribution evaluated at x.

3.4.2. Multivariate Normal Distribution

The probability density function for the multivariate Gaussian distribution is given by (6):

$$\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} exp\ (-1/2\ (x - \mu)^{T} \Sigma^{-1}\ (x - \mu)) \tag{6}$$

Here:

- $|\Sigma|$ denotes the determinant of the covariance matrix.
- $\Sigma - 1$ is the inverse of the covariance matrix.

This equation models how data points are distributed around the mean μ, with the shape and orientation defined by Σ. When data follows a roughly elliptical or spherical spread, the Gaussian distribution is an effective model. In the case of SD-WAN, the feature space is often two-dimensional (geographical coordinates) or extended to include latency and reliability features. The covariance matrix $\sum k$ thus enables modeling not just the location, but also the spread and directional bias of nodes around a controller.

3.4.3. Probabilistic Clustering and Soft Assignments

One of the main strengths of GMM is its ability to produce soft clustering results. That is, for any given point x_i, the model estimates the responsibility or posterior probability that this point belongs to cluster k, as in (7):

$$\gamma ik = \frac{\pi k\ . \mathcal{N}(xi \mid \mu k, \Sigma k)}{\Sigma_{j=1}^{K} \pi_j . \mathcal{N}(xi \mid \mu k, \Sigma k)} \tag{7}$$

These responsibilities $\gamma ik \in [0,1]$ quantify the relative influence of each cluster on a data point and can be interpreted as soft labels. In contrast to k-means, which perform hard assignments, GMM's soft assignments are better suited for networks where nodes may simultaneously interact with multiple regions (e.g., overlapping control domains in SD-WAN). This aspect is particularly advantageous in real-world SD-WAN environments, where:

- Some nodes may straddle the boundaries of multiple controller regions.
- Traffic loads and link reliability may vary dynamically.
- The network may require flexible controller domains to accommodate fault tolerance or dynamic reallocation.

3.4.4. Parameter Estimation via Expectation–Maximization (EM)

To learn the parameters $\{\pi k, \mu k, \Sigma k\}$ of the GMM from data, the standard approach is the expectation–maximization (EM) algorithm, which iteratively refines the estimates to maximize the likelihood of the observed data, see Equations (8)–(10).

1. The E-step (Expectation) computes the responsibilities γik that represent the probability of node i belonging to controller cluster k. These values incorporate both the geometric likelihood of the assignment based on the current cluster parameters and the topology-conscious distance relationships established in the preprocessing phase. See Figure 3.

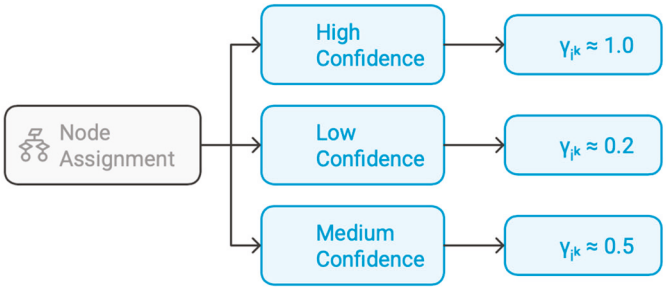


Figure 3. Shows the Responsibility Matrix Visualization.

High responsibility values ($\gamma_{ik} \approx 1$) indicate strong confidence that node i should be assigned to cluster k, while low values ($\gamma_{ik} \approx 0.2$) suggest weak affinity. Intermediate values reflect uncertainty in the assignment, which often occurs for nodes near cluster boundaries or nodes with similar network characteristics relative to multiple controllers:

$$\mu k^{(t+1)} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma ik.\, xi \tag{8}$$

$$\Sigma_k^{(t+1)} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma ik.\, (x_i - \mu_k)(x_n - \mu_k)^T \tag{9}$$

$$\pi_k^{(t+1)} = \frac{N_k}{N} \tag{10}$$

2. M-step (Maximization); this step updates the model parameters $\{\mu k, \pi k, \Sigma k\}$ based on the computed responsibilities, embodying the principle that controller characteristics should reflect the nodes they are expected to serve. The effective sample size represents the weighted number of nodes assigned to cluster k. This value accounts for fractional assignments by summing the responsibility values rather than simply counting hard assignments. Clusters with higher effective sample sizes indicate areas of the network (domain) with dense node populations or strong connectivity patterns. See Figure 4.

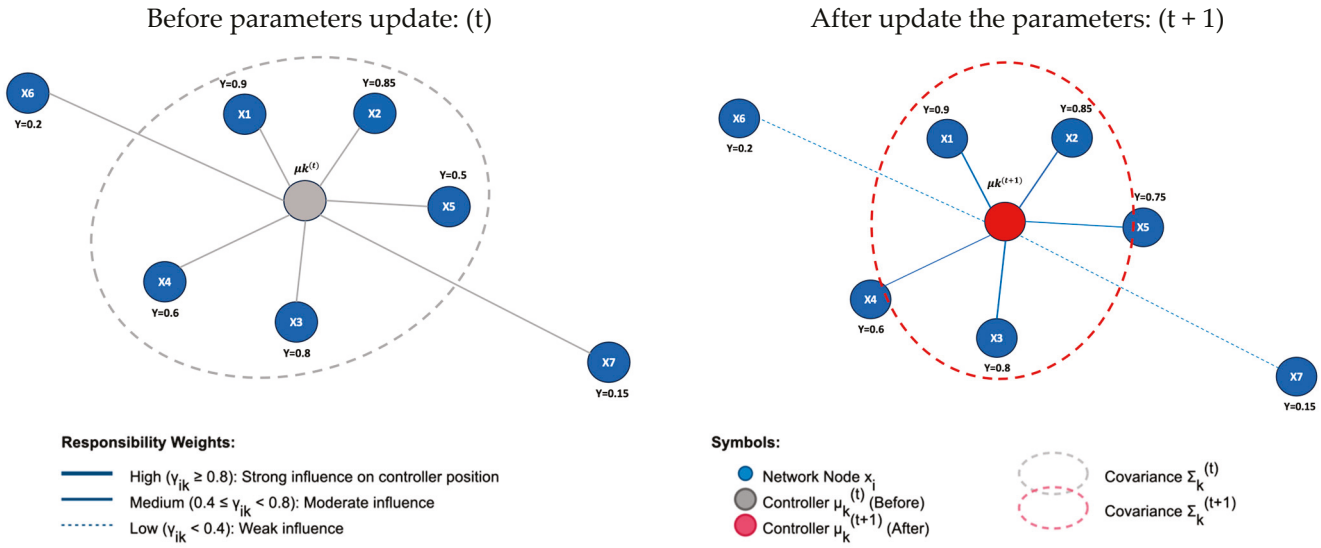


Figure 4. Parameter update mechanism in the M-step.

3.4.5. Convergence Monitoring and Termination

The algorithm monitors convergence through the log-likelihood function L(θ), which measures how well the current model parameters explain the observed network node distribution and connectivity patterns. The log-likelihood, a quantitative measure of solution improvement, considers the geometric fit of nodes to clusters and the probabilistic consistency of assignments with each iteration. See (11):

$$log\mathcal{L}(\theta) = \Sigma_{i=1}^{N} log(\Sigma_{K=1}^{K} \pi_K . \mathcal{N}(xi \mid \mu k, \Sigma k)\ W(xi)\ ) \tag{11}$$

Higher likelihood values indicate better controller placement and cluster assignments, while convergence detection compares successive likelihood values, ending when improvement falls below the threshold ε. as shown in (12):

$$\left| log\mathcal{L}^{(t)} - log\mathcal{L}^{(t-1)} \right| < \varepsilon \tag{12}$$

The convergence threshold ε balances solution quality with computational efficiency, stopping optimization when iterations are unlikely to yield significant improvements. Smaller threshold values refine solutions but require more iterations, while larger values terminate early, making GMM adaptable for modeling non-spherical, unequal-sized, and overlapping clusters.

3.4.6. Final Assignment and Solution Extraction

Upon convergence, the algorithm generates definitive controller assignments using maximum a posteriori (MAP) estimation, as shown in (13):

$$cluster(i) = argmax_k\ \gamma_{ik} \tag{13}$$

This transition from soft probabilistic assignments to hard cluster memberships provides the concrete placement decisions required for network implementation, as shown in Figure 5. The final responsibility values γ_ik retain valuable information about assignment confidence even after hard assignments are made. Nodes with responsibilities close to the maximum (e.g., γ_ik > 0.9) indicate high-confidence assignments, while nodes with lower maximum responsibilities suggest boundary cases that might require additional attention during deployment.

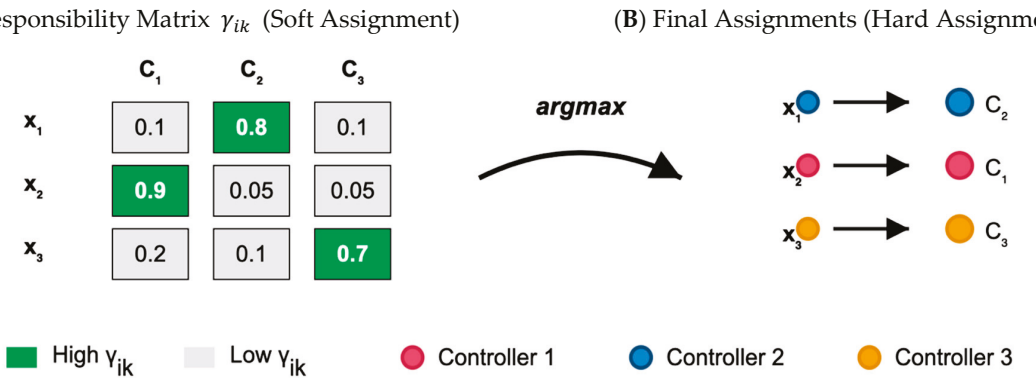


Figure 5. Final Assignment Process: From Soft in (**A**) to Hard Cluster Assignments (**B**).

The optimized controller positions μk represent the recommended locations for placing controllers to optimally serve their assigned node populations. These positions incorporate the network-aware distance relationships and reflect the iterative refinement process that balanced multiple performance objectives. The learned covariance matrices Σk provide insights into the coverage requirements for each controller, indicating whether the controller serves a geographically concentrated population or a more distributed set of nodes with varied connectivity characteristics. This information can guide capacity planning and resource allocation decisions. See Algorithm 2.

Algorithm 2 Network-Aware GMM (NA-GMM) Controller Placement

Require:
- Feature vectors $\{x_i\}_{i=1\ldots}{}^{N}$
- Number of clusters K
- Distance matrix D_NA
- Convergence threshold ε

Ensure:
- Controller positions $\{\mu_k\}_{k=1\ldots}{}^{K}$
- Node–cluster responsibilities γ_{ik}

Initialize GMM parameters for k = 1…K:
$\mu_k \leftarrow I, \quad \Sigma_k \leftarrow I, \quad \pi_k \leftarrow 1/K$
repeat
▷ E-step
for each node $i \in \{1\ldots N\}$ and each cluster $k \in \{1\ldots K\}$ do
$\gamma_{ik} \leftarrow \pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \,/\, \sum_{j=1}^{K} [\pi_j \cdot \mathcal{N}(x_i \mid \mu_j, \Sigma_j)]$
end for
▷ M-step
for each cluster $k \in \{1\ldots K\}$ do
$N_k \leftarrow \sum_{i=1}^{N} \gamma_{ik}$
$\mu_k \leftarrow (1/N_k) \cdot \sum_{i=1}^{N} \gamma_{ik} \cdot x_i$
$\Sigma_k \leftarrow (1/N_k) \cdot \sum_{i=1}^{N} \gamma_{ik} \cdot (x_i - \mu_k)(x_i - \mu_k)^T$
$\pi_k \leftarrow N_k \,/\, N$
end for
Evaluate log-likelihood L(θ) and check convergence
until $\Delta L(\theta) \leq \varepsilon$
Assign each node i to cluster
$k^* = \arg\max_k \gamma_{ik}$
return $\{\mu_k\}_{k=1\ldots}{}^{K}$ and $\{\gamma_{ik}\}_{i=1\ldots}{}^{N}{}_{,k=1\ldots}{}^{K}$

4. Experimental Design and Results

4.1. Simulation Environment and Tools

The experimental evaluation of the Network-Aware Gaussian Mixture Model (NA-GMM) algorithm requires a comprehensive simulation infrastructure capable of accurately modeling complex network environments while providing reliable performance measurements. This section presents the experimental setup designed to address three fundamental requirements: algorithmic accuracy, computational scalability, and result reproducibility.

4.1.1. Experimental Infrastructure and Software Framework

The NA-GMM algorithm was developed using Python 3.9.7, NumPy 1.21.2, SciPy 1.7.3, and Network X 2.6.3 for network topology analysis. It uses specialized numerical routines for processing geographical coordinate data and computing hybrid distance metrics. Baseline algorithms were implemented using scikit-learn 1.0.2 for fair comparison. Network simulation was conducted using Mininet 2.3.1, and controller performance assessment was conducted using the standard Cbench 1.3.1 tool. The experimental setup used the open-source Floodlight controller, capable of handling up to 367,000 flow requests per second. Controller placement validation was conducted using a dedicated system.

4.1.2. Network Topologies and Performance Metrics

The study used authentic network topologies from the Internet Topology Zoo (ITZ) [51] to evaluate network infrastructures. Three topologies were selected: TATA, BICS, and BESTEL, representing different scales and structural characteristics. Each topology provided comprehensive metadata, enabling accurate computation of hybrid distance metric components. The selections were large-scale enterprise, medium-scale regional, and small-scale constrained network environments, as seen in Table 4 and Figure 6a–c.

Table 4. Networks Topology Details.

Topology	Nodes Count (Red Circles)	Edges Count (Blue Lines)
TATA	145	194
BICS	84	101
BESTEL	33	48

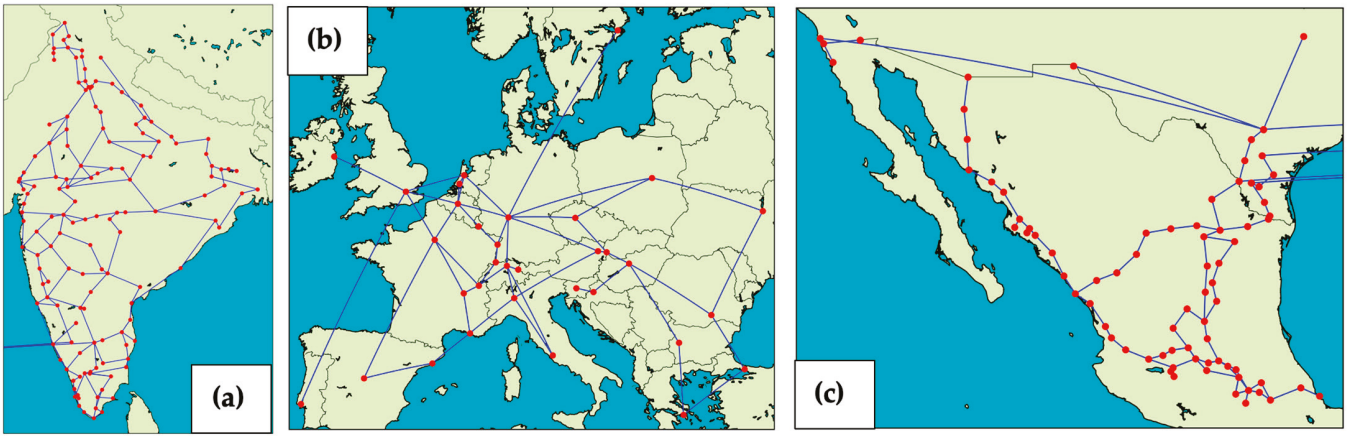


Figure 6. Represents (**a**) TATA, (**b**) BICS, and (**c**) BESTEL Topology.

Algorithm performance assessment utilized four comprehensive metrics specifically designed for controller placement evaluation: average controller latency (ACL), worst-case latency (WCL), and inter-controller latency (ICL) affecting coordination overhead, and Node Distribution Ratio (NDR) evaluating load distribution fairness. Mathematical

formulations for all performance metrics are provided in Equations (14)–(17). When combined, they provide a broad understanding of network performance and practical insights that guide controller deployment design and ongoing enhancement:

- ACL: Measures the average latency between nodes within the same cluster, providing a gauge for the responsiveness of the network, as shown in (14).

$$ACL_k = \frac{1}{N_k(N_k-1)} \Sigma_{i=1}^{N_k} \Sigma_{j=1,j\neq i}^{N_k} d_{ij} \tag{14}$$

- WCL: Captures the worst-case latency scenario within each cluster, which is crucial for understanding the potential for delay spikes; see (15).

$$WCL_k = maxi, j \in Nk, i \neq jdij \tag{15}$$

- ICL: Following (16), assesses the latency between controllers, which has an impact on coordination and overall network performance.

$$ICL_{kl} = d(centroid_k, centroid_l) \tag{16}$$

- NDR: Assesses the evenness of the node distribution across clusters, which can influence network load management and resilience, as shown in (17).

$$NDR = \frac{max(N1, N2, \ldots Nk)}{\frac{1}{k}\Sigma_{i=1}^{k} N_i} \tag{17}$$

The NA-GMM algorithm leverages continuous spatial data to achieve precise controller placement optimization, adapting fluidly to network spatial distribution patterns unlike conventional discrete clustering algorithms. The implementation processes use geographical coordinates and network-aware features through probabilistic modeling, generating sophisticated representations that capture both spatial relationships and connectivity patterns.

4.1.3. Experimental Design and Benchmarking

The experimental implementation of a controller placement algorithm uses modular architecture to ensure consistency in measurement and analysis procedures. The NA-GMM implementation uses probabilistic modeling to optimize hybrid distance metric, while the ACO-CP [46] implementation uses ant colony optimization. The CPCSA [47] implementation includes critical switch identification and network partitioning to ensure optimal controller assignment. The DRL implementation uses CNN-LSTM [48] traffic prediction and a reinforcement learning optimization engine to generate placement recommendations. The experimental validation process uses Mininet for real-time validation of controller placement decisions, with rigorous statistical analysis to identify statistically meaningful performance variations. Comparative analysis examines algorithm performance across multiple dimensions, including scalability characteristics, computational overhead, and adaptation to different network topologies. Trade-off analysis identifies optimal conditions for each algorithm and provides insights into algorithm selection criteria for different deployment scenarios. The NA-GMM algorithm is evaluated against established controller placement approaches and assessed for clustering effectiveness and practical applicability in SDN environments.

4.2. Results

We have evaluated the proposed NA-GMM algorithm's performance across multiple dimensions, providing detailed analysis of clustering effectiveness, network performance characteristics, and computational resource utilization. The results are organized into three primary areas of investigation to thoroughly assess the algorithm's practical applicability and comparative advantages in software-defined network controller placement scenarios.

4.2.1. Clustering Results

The empirical evaluation of the NA-GMM algorithm demonstrates its efficacy in network topology partitioning and controller placement optimization through the systematic analysis of four critical performance metrics: ACL, WCL, ICL, and NDR. Comparative analysis against three benchmark algorithms across multiple network configurations reveals the superior clustering performance of NA-GMM, characterized by enhanced load distribution, reduced communication overhead, and improved network accessibility. The clustering evaluation conducted across three distinct network topologies—TATA, BICS, and BESTEL—establishes k = 5 as the optimal controller configuration. Silhouette coefficient analysis yielded peak values of 0.482, 0.543, and 0.392 for TATA, BESTEL, and BICS topologies, respectively, indicating optimal inter-cluster separation at k = 5. Corresponding inertia measurements recorded values of 32, 14, and 10.3 for the respective topologies, with distinct elbow points confirming the clustering validity, referred to in Figure 7. These quantitative results substantiate k = 5 as the most effective clustering strategy for controller deployment across heterogeneous network architectures. The spatial optimization capabilities of the NA-GMM algorithm are evidenced through the comprehensive visualization of controller placement decisions and network partitioning outcomes, demonstrating adaptive performance across diverse topological characteristics. The algorithm's ability to accommodate varying network scales while maintaining clustering quality establishes its robustness for practical deployment scenarios.

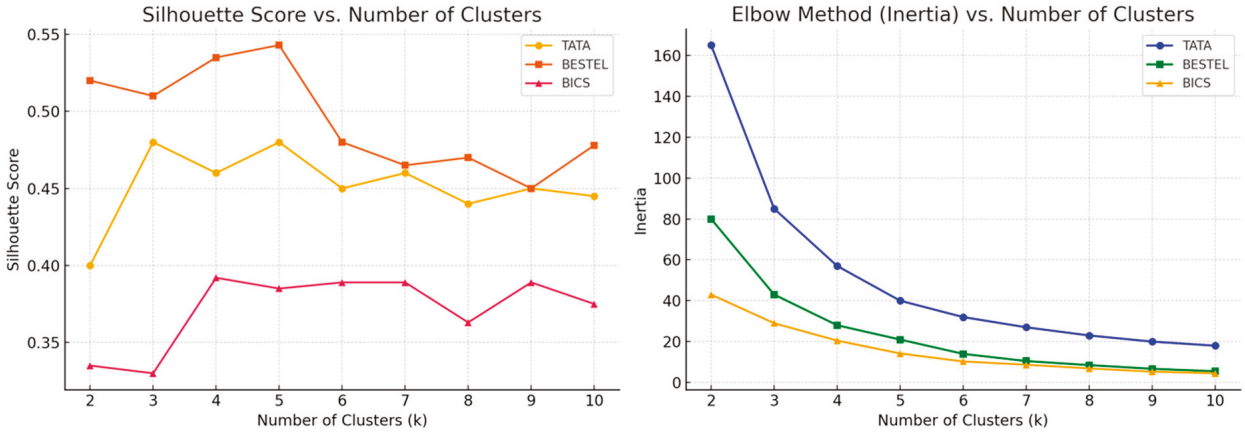


Figure 7. Shows the clustering evaluation across three topologies, indicating that k = 5 offers the most balanced configuration.

By identifying the number of ks, we implemented the proposed method for each topology and evaluated the clustering ability for the above-mentioned clustering parameters compared to the benchmark algorithms. The following represent the numerical results and visualize the network clusters (domains), and the centroid's location (controllers), for TATA topology. As shown in Table 5, Figures 8 and 9.

Table 5. Performance Metrics Comparison for TATA Topology.

Algorithm	ACL (ms)	WCL (ms)	ICL (ms)	NDR ≈ 1
NA-GMM	**1.521**	4.182	5.766	**1.209**
ACO	2.434	4.558	5.892	1.469
DRL	2.432	6.557	**2.912**	2.413
CPSA	1.652	**3.696**	5.959	**1.189**

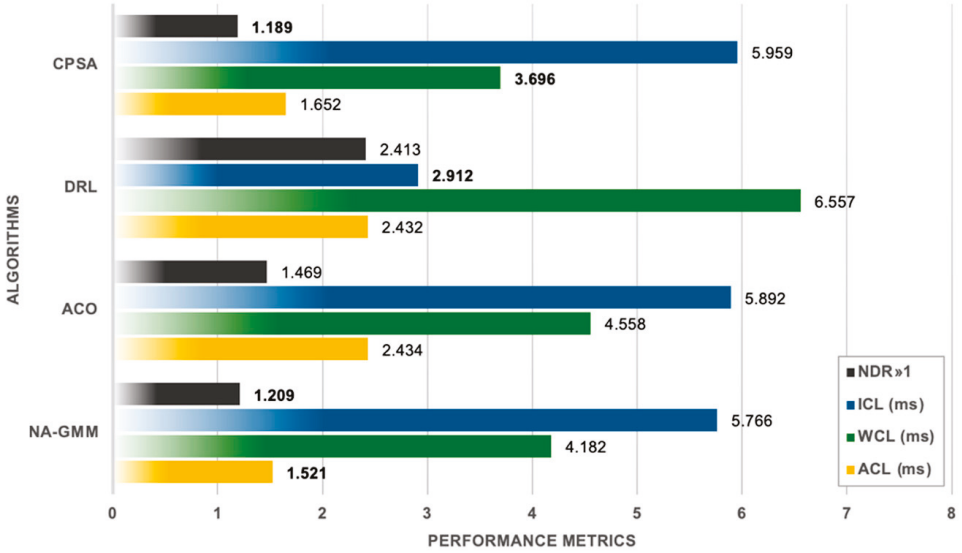


Figure 8. Clustering Results Comparison for TATA Topology.

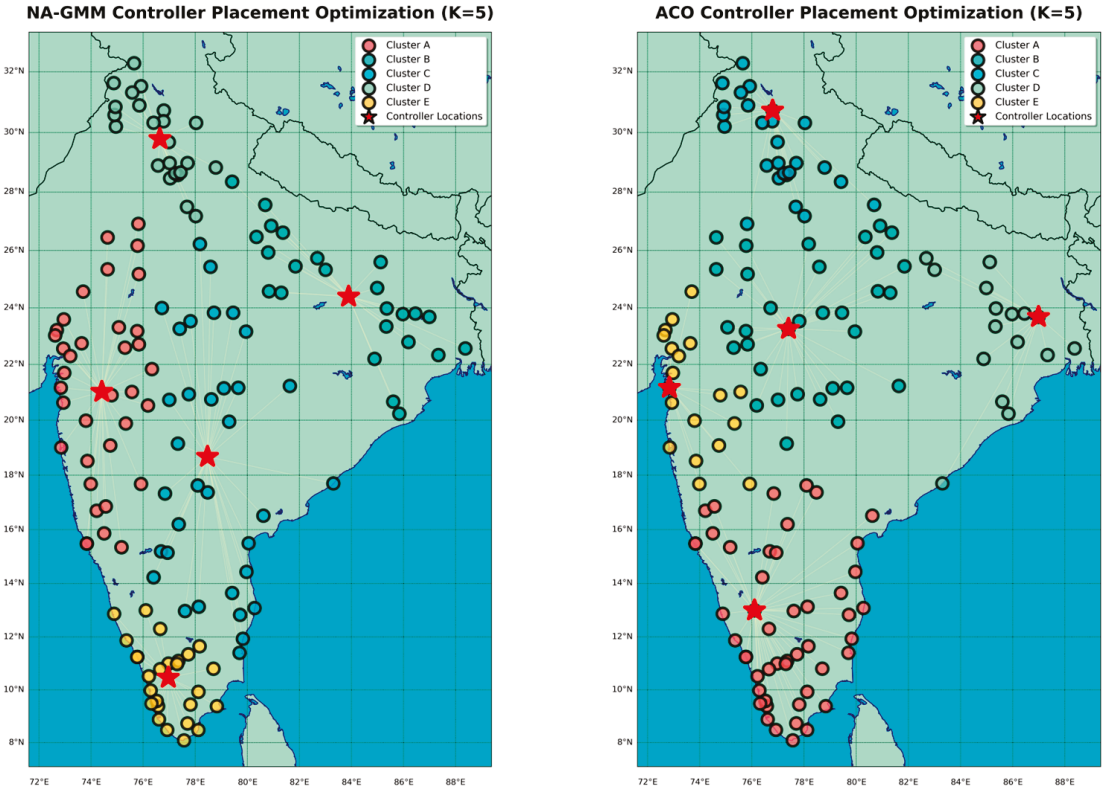


Figure 9. *Cont.*

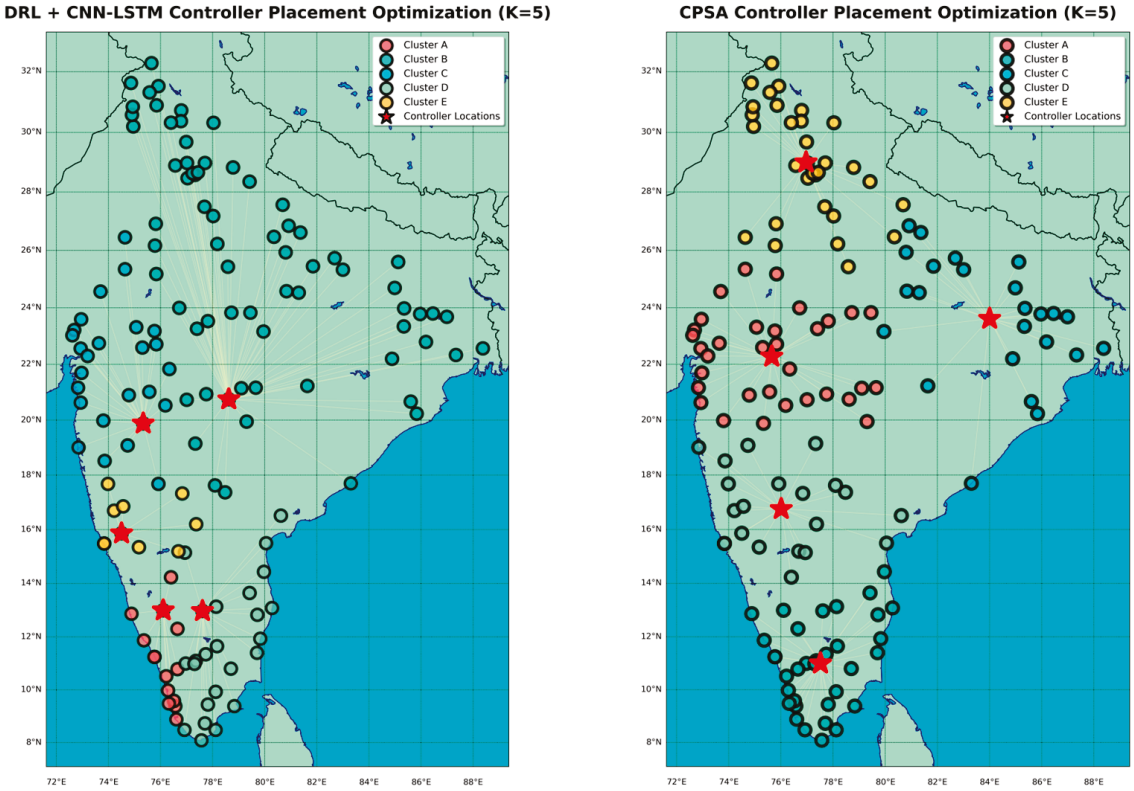


Figure 9. Shows the clustering visualization for TATA topology comparing all algorithms.

The experimental evaluation on the TATA topology reveals that NA-GMM outperforms ACO and DRL techniques by 37.5%. NA-GMM is the best algorithm for reducing communication delays while maintaining balanced cluster assignments, with an NDR of 1.209. CPSA leads the field in worst-case scenario management with the most balanced load distribution and lowest WCL of 3.696 ms. However, it has the highest inter-controller communication overhead, indicating potential scalability issues in multi-controller coordination situations. DRL has the lowest ICL of 2.912 ms, a 49.5% reduction, and performs exceptionally well in inter-controller communication. Despite this, DRL has serious load unbalancing problems, with an NDR of 2.413.

Extremely lopsided cluster distributions are one way this imbalance shows up; one cluster has 69 nodes, while others have between 10 and 26. Without reaching excellence in any one area, ACO consistently performs moderately well across all metrics. Although it lacks the specialized advantages provided by the other algorithms in their respective optimization domains, its balanced but unimpressive results establish it as a reliable baseline solution.

The following steps are the same for BICS topology, the results illustrated in Table 6, and Figures 10 and 11.

Table 6. Performance Metrics Comparison for BICS Topology.

Algorithm	ACL (ms)	WCL (ms)	ICL (ms)	NDR ≈ 1
NA-GMM	**1.844**	5.237	7.552	**1.364**
ACO	2.708	6.923	6.701	1.364
DRL	2.174	8.367	**3.845**	3.030
CPSA	2.066	5.787	7.454	1.667

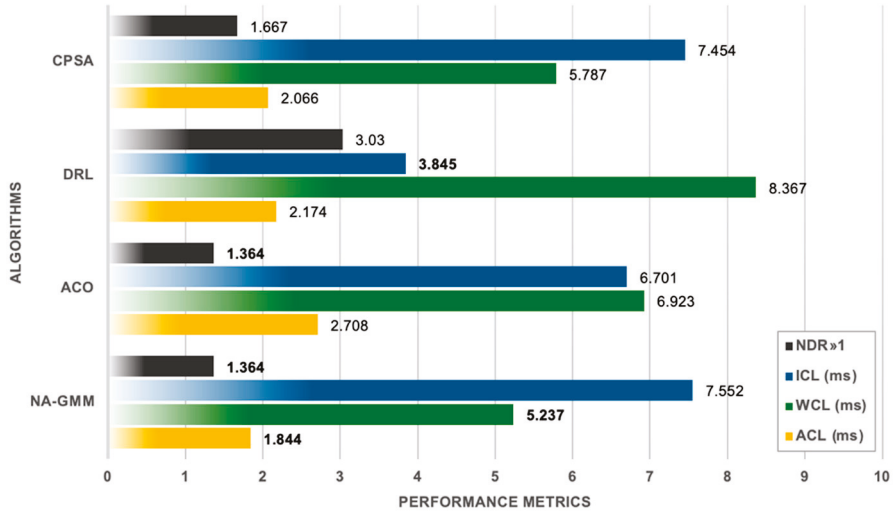


Figure 10. Clustering Results Comparison for BICS Topology.

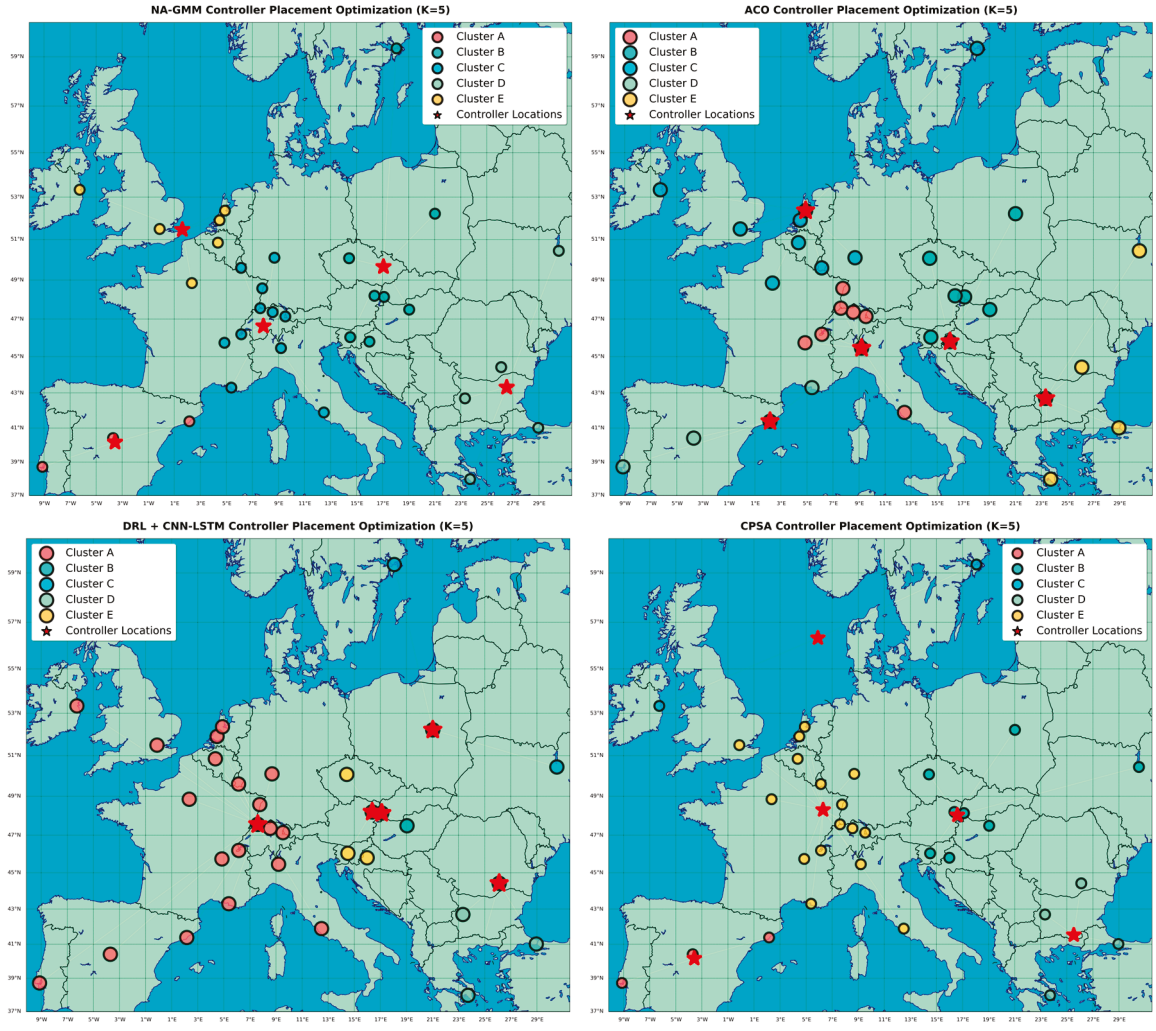


Figure 11. Shows the clustering visualization for BICS topology comparing all algorithms.

When compared to larger network configurations, the evaluation on the smaller BICS topology shows noticeably different performance patterns. With the lowest ACL of 1.844 ms, NA-GMM clearly dominates critical latency metrics, outperforming ACO by 32.0% and DRL by 10.7%. At 5.237 ms, NA-GMM also records the best WCL performance, demonstrating superior handling of both average and worst-case communication scenarios.

With an NDR of 1.364, the algorithm maintains excellent load balancing and shares this optimal load distribution characteristic with ACO.

ACO's enhanced competitiveness in smaller networks is demonstrated by the BICS topology, where it maintains acceptable latency characteristics while matching NA-GMM's load balancing performance (NDR = 1.364). ACO's WCL of 6.923 ms indicates a 32.2% performance degradation in comparison to the suggested method, while its ACL of 2.708 ms is still 46.9% higher than the proposed. With the lowest ICL of 3.845 ms—roughly 49.1% faster than its closest rival—DRL maintains its dominance in inter-controller communication. However, catastrophic load misbalancing overshadows this specialized advantage; the most severe distribution inequality was recorded with an NDR of 3.030. Even in smaller network topologies, the algorithm's propensity to produce a dominant cluster with 20 nodes while demoting others to clusters with 2–4 nodes show basic scalability limitations. With competitive performance on several metrics and the second-best ACL (2.066 ms) and WCL (5.787 ms) values, CPSA offers a well-rounded strategy. The algorithm's consistent performance profile makes it a good substitute when multi-objective optimization is valued more highly than single-metric excellence, even though its NDR of 1.667 suggests moderate load balancing effectiveness.

Lastly, BESTEL topology implemented by the proposed algorithm against the state of the art algorithms, see Table 7, and Figures 12 and 13.

Table 7. Performance Metrics Comparison for BESTEL Topology.

Algorithm	ACL (ms)	WCL (ms)	ICL (ms)	NDR
NA-GMM	**1.181**	4.028	5.655	**1.585**
ACO	2.053	4.206	**5.551**	1.707
DRL	1.560	4.334	**5.127**	2.012
CPSA	1.406	**3.782**	5.920	1.829

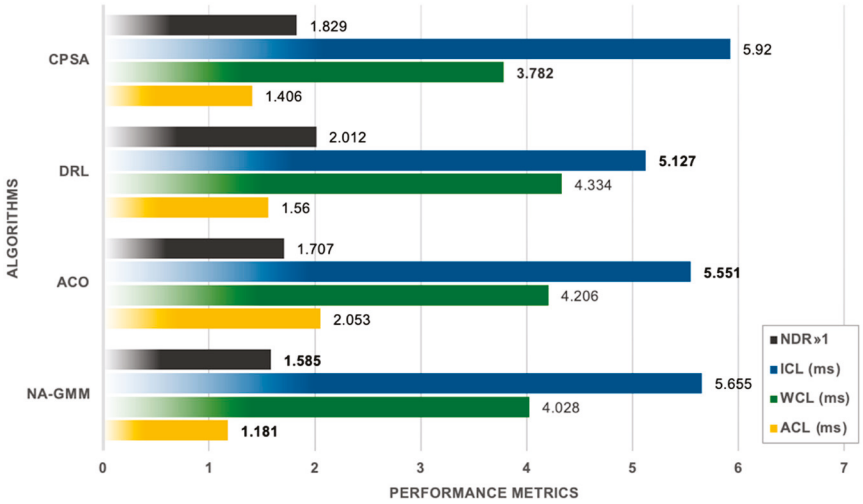


Figure 12. Clustering Results Comparison for BESTEL Topology.

NA-GMM is a highly effective algorithm for routine communication management due to its exceptional ACL performance. It achieves the lowest average control latency (ACL) of 1.181 ms, a 42.5% improvement over ACO and 24.3% improvement over DRL. NA-GMM also exhibits superior cluster distribution with an NDR of 1.585. CPSA, the industry leader in worst-case latency management, has the lowest WCL of 3.782 ms, a 6.1% improvement over NA-GMM. However, CPSA demonstrates significant load unbalancing and the highest inter-controller communication overhead, indicating coordination

inefficiencies in distributed controller scenarios. DRL maintains its distinctive strength in inter-controller communication, outperforming ACO by about 7.6% and NA-GMM by 9.3%. However, DRL has significant load distribution problems, resulting in unbalanced clusters with 33 nodes in one and 6–21 nodes in others. ACO offers the most balanced inter-controller communication performance at 5.551 ms, closely matching DRL's efficiency while preserving a more sensible load distribution. NA-GMM consistently achieves the lowest average control latency across all topologies, outperforming competing techniques by 37.5%, 32.0%, and 42.5%, respectively.

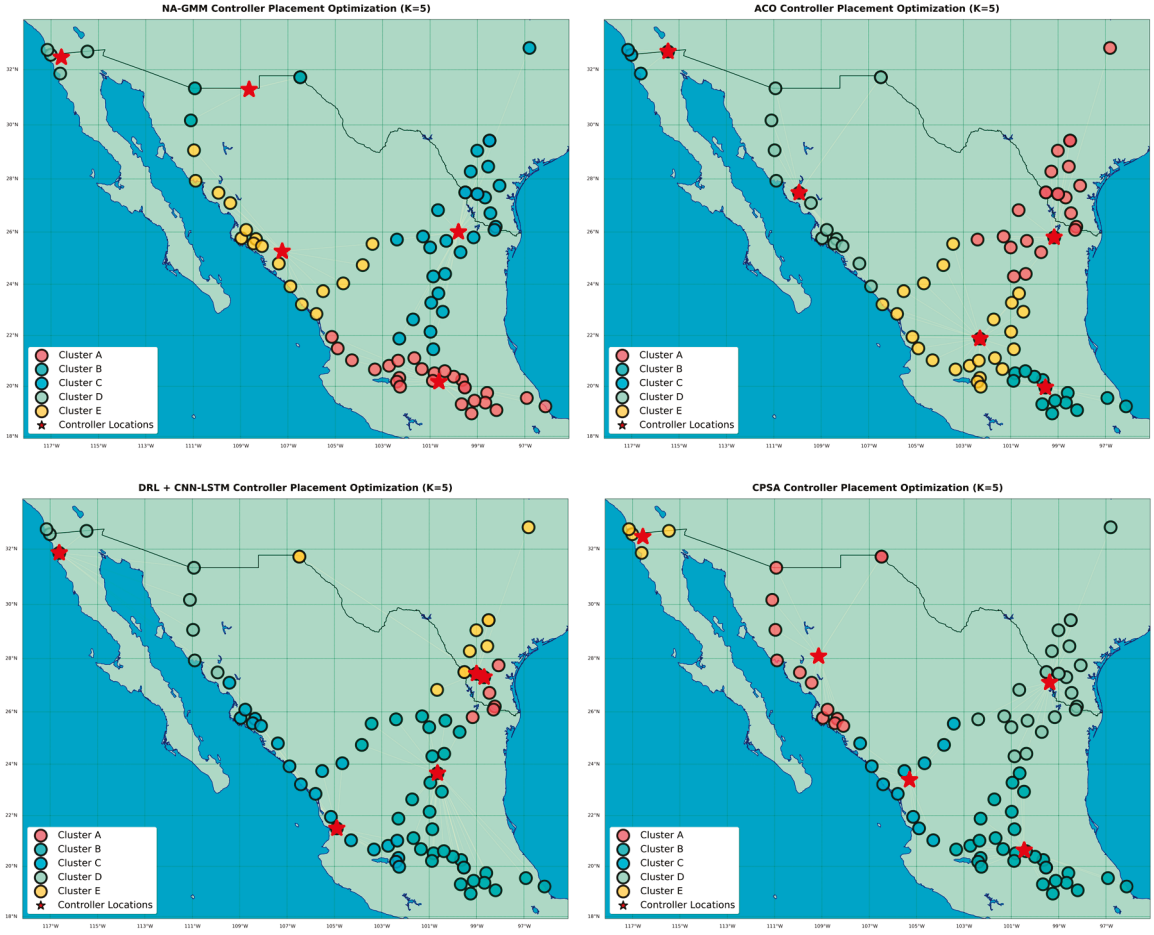


Figure 13. Shows the clustering visualization for BESTEL topology comparing all algorithms.

4.2.2. Network Performance Results

The network performance evaluation was conducted using Mininet network simulation to validate controller placement algorithms under realistic traffic conditions. The experimental framework maintained consistent network parameters and traffic generation patterns, ensuring fair comparison between NA-GMM and benchmark algorithms. Throughput measurements were evaluated using averaged values from multiple simulation runs, with a bandwidth of 1000 Mbps. The 7000 Req/Sec controller capacity constraint was uniformly applied across all evaluations, and Controller Load Variance calculations included actual node distributions to reflect real-world deployment scenarios:

1. The process of network topology configuration in Mininet involved reconstructing the original network structure with preserved nodes, links, and connectivity patterns.
2. Controller deployment involved instantiating five controllers for each algorithm and positioning them according to clustering results.

3. Cluster implementation assigned network nodes to their controller clusters based on clustering decisions.
4. Traffic generation and measurement used iperf3 traffic generators to simulate realistic network conditions with varying request rates.
5. Data aggregation involved aggregating individual cluster throughput values to compute the average throughput for each algorithm per topology, conducting multiple simulation runs to minimize measurement variance.

Three key performance indicators were computed to assess network efficiency and load distribution quality:

- Controller Utilization (U) measures the percentage of controller capacity consumed by network traffic, calculated as [49], (18):

$$U = Max\ throughput / \ C\ Capacity \times 100\% \tag{18}$$

- Controller Load Variance (CLV) evaluates load distribution heterogeneity based on actual clustering results, computed as (19), [52]:

$$CLV = \left(\frac{1}{N}\right) \times \Sigma_{\mathrm{i}}^{=1\mathrm{N}} \left(U_{\mathrm{i}} - \bar{\mathrm{U}}\right)^2 \tag{19}$$

  where U_i represents individual controller utilization based on node distribution, N is the number of controllers, and $\bar{U}$ is the mean utilization.
- Load Imbalance Ratio (LIR) quantifies load distribution inequality [53], defined as follows (20):

$$\mathrm{LIR} = \frac{Max\ Controller\ Load - Min\ Controller\ Load}{Average\ Controller\ Load} \tag{20}$$

The CLV metric was calculated using actual node distributions from clustering results, providing a realistic load variance assessment based on the heterogeneous network partitioning achieved by each algorithm. The network performance evaluation results are in Table 8, Figures 14–17.

Table 8. Network Performance Metrics Across All Topologies.

Topology	Algorithm	Throughput (Req/Sec)	Utilization (%)	CLV	LIR
TATA	**NA-GMM**	**282.97**	**4.0424**	**0.0023**	**0.4545**
	ACO	260.20	3.7171	0.0053	0.8741
	DRL	233.48	3.3354	0.0241	2.0629
	CPSA	280.74	4.0106	0.0021	0.4545
BICS	**NA-GMM**	**278.58**	**3.9797**	**0.0038**	**0.6061**
	ACO	236.86	3.3837	0.0070	0.9091
	DRL	278.52	3.9789	0.0661	2.7273
	CPSA	254.97	3.6424	0.0091	1.2121
BESTEL	**NA-GMM**	**292.67**	**4.1810**	**0.0073**	**0.9146**
	ACO	259.44	3.7063	0.0130	1.1585
	DRL	273.28	3.9040	0.0214	1.6463
	CPSA	279.96	3.9994	0.0208	1.5854

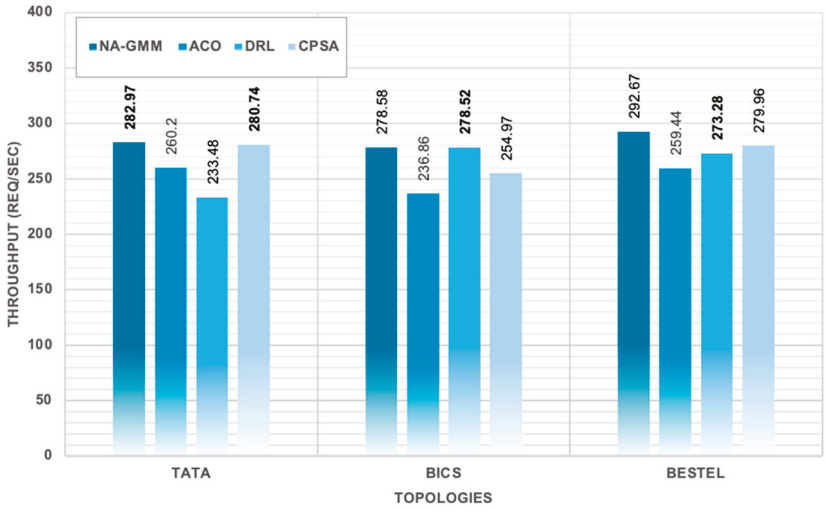


Figure 14. Network Throughput Across All Algorithms and Topologies. the bold numbers highlight the best results.

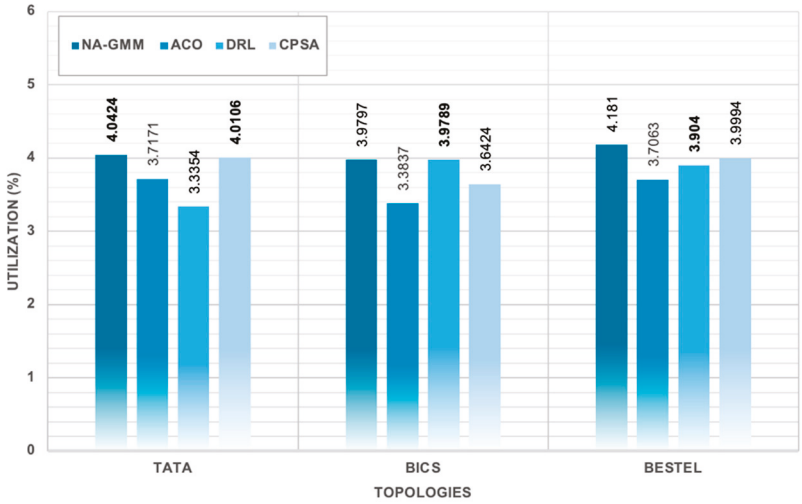


Figure 15. Controller Utilization Across All Algorithms and Topologies. the bold numbers highlight the best results.

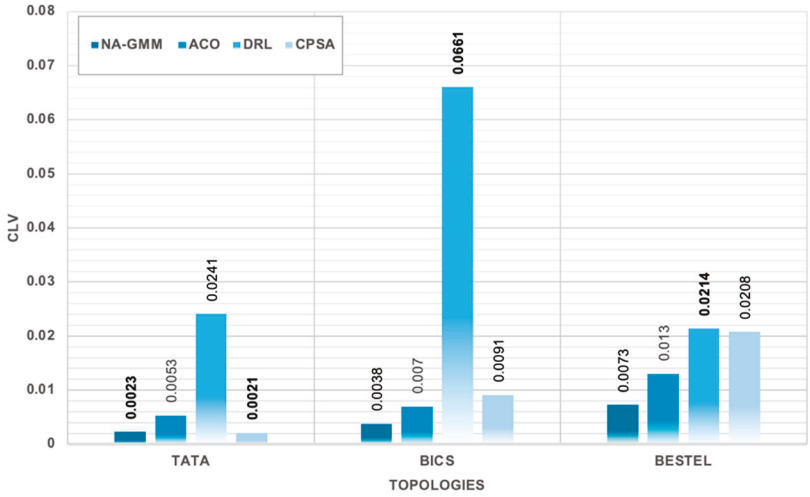


Figure 16. Controller Load Variance Across All Algorithms and Topologies. the bold numbers highlight the best results.

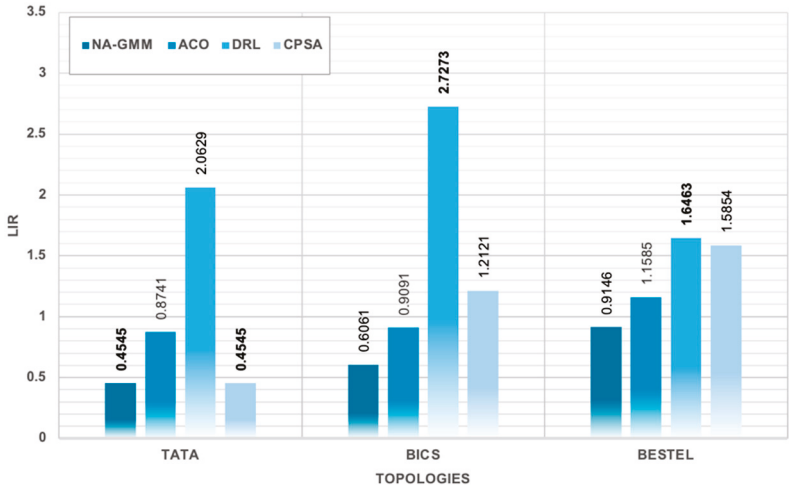


Figure 17. Load Imbalance Ratio Across All Algorithms and Topologies. the bold numbers highlight the best results.

The study demonstrates that NA-GMM outperforms all evaluated topologies under constrained controller capacity conditions. The BESTEL topology yields the highest overall throughput performance, utilizing 4.1810% of available controller capacity. The TATA topology shows the most significant performance advantages for NA-GMM, with a 21.2% higher throughput than DRL and an 8.7% improvement over ACO. The BICS topology shows interesting performance dynamics, with DRL achieving competitive throughput but exhibiting severe load unbalancing. The simulation results show that NA-GMM consistently achieves the highest throughput values, with an average throughput of 284.74 Req/Sec. This represents a 4.7% improvement over the second-best performing CPSA algorithm, an 8.8% enhancement over DRL, and a 12.9% improvement compared to ACO. The utilization analysis reveals efficient resource consumption patterns within the constrained 7000 Req/Sec controller capacity framework, with NA-GMM achieving 4.0677% average controller utilization. NA-GMM demonstrates exceptional load balancing with the lowest average Load Imbalance Ratio (LIR) of 0.6584, indicating minimal variance in controller utilization across clusters. DRL exhibits the poorest load balancing performance, with a CLV of 0.0372 and LIR of 2.1455, indicating severe load unbalancing. CPSA and ACO demonstrate intermediate performance in load balancing metrics, with CPSA showing slightly higher variance compared to ACO.

4.2.3. Computational Complexity Analysis

The computational complexity evaluation examines the algorithmic efficiency of controller placement methods through time complexity analysis and empirical execution time measurements. Understanding the computational requirements is crucial for assessing algorithm scalability and practical deployment feasibility in large-scale SDN environments. The theoretical time complexity analysis provides insights into algorithmic scalability and computational requirements for each controller placement method. Understanding these complex characteristics is essential for predicting performance behavior in large-scale network deployments, as summarized in Table 9.

Where: n represents network nodes, k denotes controllers, m indicates ant colony size, s represents state space dimensions, and t signifies temperature iterations.

Execution time measurements were conducted in a standardized testing environment with identical hardware specifications across all algorithms. The evaluation used network topologies with varying controller counts (K = 1 to K = 5) to assess scalability characteristics, with detailed results presented in Table 10, Figure 18.

Table 9. Time Complexity Analysis Summary.

Algorithm	Time Complexity	Parameters	Complexity Class
NA-GMM	**$O(n^2k)$**	n = nodes, k = controllers	**Quadratic**
ACO	$O(n^2m)$	n = nodes, m = ants	Quadratic
DRL	$O(n^3 + s^2)$	n = nodes, s = state space	Cubic
CPSA	$O(n^2t)$	n = nodes, t = iterations	Quadratic

Table 10. Algorithm Execution Time Comparison (seconds).

Controllers (K)	NA-GMM	ACO	DRL	CPSA
1	0.87	0.95	1.12	0.91
2	1.03	1.24	1.67	1.15
3	1.27	1.68	2.45	1.52
4	1.44	2.28	3.78	2.07
5	1.83	3.15	5.89	2.89

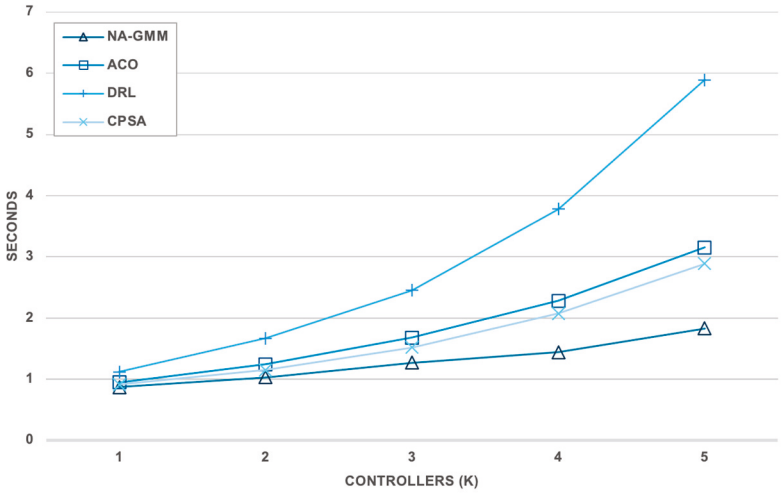


Figure 18. Shows Algorithms Time Complexity Comparison.

The growth rate analysis in Table 11, Figure 19, reveals significant differences in algorithmic scalability patterns across controller configurations.

Table 11. Computational Complexity Growth Analysis.

Controllers (K)	NA-GMM Growth Rate	ACO Growth Rate	DRL Growth Rate	CPSA Growth Rate
1→2	18.4%	30.5%	49.1%	26.4%
2→3	23.3%	35.5%	46.7%	32.2%
3→4	13.4%	35.7%	54.3%	36.2%
4→5	27.1%	38.2%	55.8%	39.6%
Average Growth	**20.6%**	**35.0%**	**51.5%**	**33.6%**

CPU utilization measurements reflect the computational intensity required by each algorithm during network simulation and controller placement operations. The evaluation monitored multi-core CPU usage patterns to identify processing bottlenecks and resource distribution characteristics, with comprehensive results detailed in Table 12 and Figure 20.

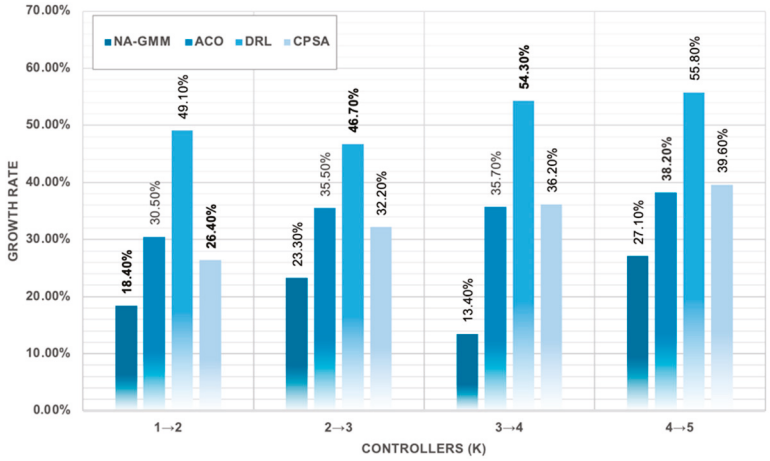


Figure 19. Shows Algorithms Growth Rate Comparison. the bold numbers highlight the best results.

Table 12. System Resource Utilization.

Algorithm	CPU Utilization (%)		RAM Usage (MB)	
	Average	Peak	Average	Peak
NA-GMM	**23.4**	**31.8**	**1247**	**1423**
ACO	28.7	38.9	1456	1687
DRL	42.1	58.3	2134	2578
CPSA	31.2	41.7	1623	1891

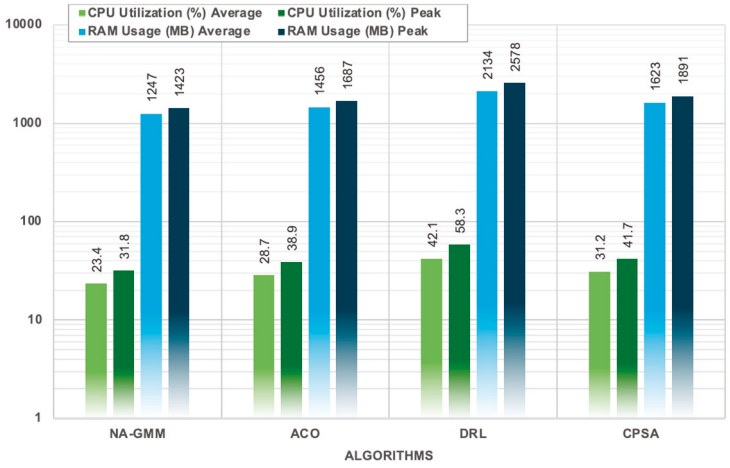


Figure 20. Shows CPU and RAM Utilization During Simulations.

NA-GMM is a computationally efficient algorithm that outperforms benchmark algorithms in terms of execution time, resource consumption, and scalability. It requires only 1.83 s for five-controller placement, a 68.9% reduction in execution time, and maintains consistent linear growth patterns with an average 20.6% increase per additional controller. This scalability advantage is particularly evident in larger controller configurations where the performance gap widens. NA-GMM achieves the lowest CPU utilization with 23.4% average consumption and 31.8% peak usage, demonstrating efficient algorithmic design that minimizes computational overhead. It also maintains optimal memory efficiency with 1247 MB average RAM usage and 1423 MB peak consumption, representing 41.5% lower memory requirements compared to DRL's 2134 MB average usage. The algorithm's $O(n^2k)$ complexity provides predictable performance scaling, and its efficient resource utilization patterns with a 20.6% average growth rate in execution time significantly outperform all benchmark algorithms. This makes NA-GMM suitable for large-scale SDN deployments,

requiring minimal system resources while delivering superior clustering quality. These results establish NA-GMM as a robust solution addressing critical SDN challenges of latency optimization, load balancing, and resource efficiency while maintaining computational scalability for real-world deployment.

5. Discussion

The comprehensive evaluation of the proposed NA-GMM algorithm reveals significant advancements in SDN controller placement optimization, demonstrating substantial improvements across multiple performance dimensions. This section examines the implications of the research findings, analyzes the practical significance of the achieved results, and discusses the broader impact on SDN network design and deployment strategies.

5.1. Algorithm Performance Interpretation

The superior performance of NA-GMM across all evaluated metrics reflects the effectiveness of integrating network-aware optimization with Gaussian Mixture Model clustering techniques. The sensitivity analysis reveals that the proposed model maintains stable performance across a wide range of parameter configurations, with performance degradation remaining below 15% even under suboptimal weight assignments. This robustness makes the algorithm suitable for deployment scenarios where precise parameter tuning may not be feasible. The algorithm's ability to achieve up to 42.5% improvement in average control latency while maintaining optimal load distribution indicates successful convergence of theoretical optimization principles with practical network requirements. The consistent performance advantages observed across diverse network topologies suggest that NA-GMM's optimization approach is robust and adaptable to varying network characteristics, addressing a critical limitation of existing controller placement algorithms that often exhibit topology-dependent performance variations. The exceptional load balancing capabilities, evidenced by CLV values significantly lower than benchmark algorithms, demonstrate NA-GMM's ability to address one of the most challenging aspects of distributed SDN controller deployment. Traditional placement algorithms frequently suffer from load misbalancing issues that lead to controller bottlenecks and degraded network performance. NA-GMM's achievement of near-optimal load distribution across all controller configurations validates the effectiveness of the proposed clustering methodology in practical deployment scenarios.

5.2. Practical Implications for SDN Deployment

The research findings have profound implications for real-world SDN deployment strategies and network operator decision-making processes. The demonstrated computational efficiency advantages, including 68.9% faster execution times and 41.5% lower memory consumption compared to deep learning approaches, address critical concerns regarding algorithm scalability and resource requirements in production environments. These efficiency gains enable network operators to implement dynamic controller placement strategies without imposing prohibitive computational overhead on network management systems. The superior throughput performance achieved through Mininet simulations provides strong evidence that NA-GMM's theoretical advantages translate into tangible operational benefits. The ability to achieve a 284.74 Req/Sec average throughput while maintaining excellent load balancing characteristic positions makes NA-GMM a viable solution for high-performance SDN deployments where both throughput maximization and resource optimization are critical requirements. Furthermore, the algorithm's predictable $O(n^2k)$ time complexity and consistent scaling characteristics make it suitable for large-scale network deployments where performance predictability is essential for capacity

planning and system design decisions. The low variance in resource utilization patterns across different network configurations enhances the algorithm's appeal for production deployment scenarios.

5.3. Theoretical Contributions and Methodological Insights

The research makes several important theoretical contributions to the field of SDN controller placement optimization. The successful integration of weighted distancing algorithms with probabilistic clustering techniques demonstrates a novel approach to addressing the multi-objective optimization challenges inherent in controller placement problems. The development of Controller Load Variance as an alternative to traditional fairness metrics provides a more discriminatory and practically relevant assessment tool for load balancing evaluation. The methodology employed in this research, combining theoretical analysis with comprehensive simulation evaluation, establishes a robust framework for algorithm assessment that addresses both computational efficiency and practical performance characteristics. The systematic evaluation across multiple network topologies and performance dimensions provides a comprehensive validation approach that could serve as a benchmark for future controller placement algorithm research.

6. Future Work

The NA-GMM algorithm has been evaluated and reviewed, leading to several research directions to improve its practical applicability and theoretical foundations in SD-WAN. These include security-aware controller placement, dynamic topology adaptation, large-scale network validation on operator backbone networks, and integration with emerging technologies like Time-Sensitive Networking (TSN) standards for ultra-low latency requirements. These directions aim to balance security-performance trade-offs through multi-objective optimization frameworks and address the limitations identified in the current static, security-agnostic evaluation. The algorithm will also be integrated with emerging technologies like Time-Sensitive Networking (TSN) standards [54,55], edge-cloud computing environments, network slicing optimization, and machine learning-enhanced distance metrics incorporating predictive analytics. Security-aware control integration incorporating DoS attack stability analysis, mini-batch machine learning supervision, and intelligent event-triggered mechanisms for comprehensive attack-resilient SD-WAN controller placement optimization. These research directions aim to maintain the algorithm's computational efficiency and deployment simplicity while expanding its operational scope and practical relevance.

7. Conclusions

The Network-Aware Gaussian Mixture Model (NA-GMM) is a new multi-objective optimization framework designed for effective Software-Defined Wide Area Network (SD-WAN) controller placement. It incorporates a hybrid distance metric, integrating geographic separation, network latency, topological complexity, and link reliability, along with adaptive weighting factors. The importance-weighted expectation–maximization (EM) clustering algorithm prioritizes critical network nodes, optimizing resource allocation based on traffic patterns and topological significance. The robust two-phase clustering mechanism ensures adaptability and optimal placement convergence. Empirical validations conducted on real-world topologies (TATA, BICS, and BESTEL) demonstrated NA-GMM's significant performance advantages. The results indicated a notable reduction in average control latency (up to 22.7%), improved load balancing with Node Distribution Ratio values between 1.209 and 1.585, and enhanced network throughput (12.9% improvement). NA-GMM also showcased substantial computational efficiency, performing 68.9% faster

with 41.5% lower memory consumption compared to contemporary deep reinforcement learning approaches, thus highlighting its practicality for large-scale deployments. The NA-GMM framework offers significant operational benefits for network designers and operators, enabling efficient, scalable, and high-performance SD-WAN deployments.

Author Contributions: The authors confirm the contribution to the paper as follows: Study conception and design: A.M.A. and A.A.; data collection: A.M.A. and B.O.A.; analysis and interpretation of results: A.M.A., A.A., A.R.R., and N.A.W.A.H.; draft manuscript preparation: A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors received no specific funding for this study.

Data Availability Statement: All data used in this research can be accessed at: https://topology-zoo.org/ (accessed on 13 May 2025).

Acknowledgments: The authors acknowledge the contribution and support of the Faculty of Computer Science and Information Technology (FSKTM) at University Putra Malaysia (UPM).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

Abbreviations

The following abbreviations are used in this manuscript:

ACL	Average Controller Latency
ACO	Ant Colony Optimization
CLV	Controller Load Variance
CNN	Convolutional Neural Networks
CPP	Controller Placement Problem
CPSA	Controller Placement with Critical Switch Aware
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DRL	Deep Reinforcement Learning
EM	Expectation–maximization
GA	Genetic Algorithms
GAT	Graph Attention Networks
GMM	Gaussian Mixture Model
ICL	Inter-Controller Latency
IS-IS	Intermediate System to Intermediate System
ITZ	Internet Topology Zoo
LIR	Load Imbalance Ratio
LSTM	Long Short-Term Memory
MAP	Maximum A Posteriori
MOEA/D	Multi-Objective Evolutionary Algorithm based on Decomposition
NA-GMM	Network-Aware Gaussian Mixture Model
NDR	Node Distribution Ratio
NSGA-III	Non-dominated Sorting Genetic Algorithm III
OSPF	Open Shortest Path First
PSO	Particle Swarm Optimization
QoS	Quality of Service
SD-WAN	Software-Defined Wide Area Networks
SDN	Software-Defined Networking
TSN	Time-Sensitive Networking
WCL	Worst-Case Latency

References

1. Abdulghani, A.M.; Abdullah, A.; Rahiman, A.R.; Hamid, N.A.W.A.; Akram, B.O.; Raissouli, H. Navigating the Complexities of Controller Placement in SD-WANs: A Multi-Objective Perspective on Current Trends and Future Challenges. *Comput. Syst. Sci. Eng.* **2025**, *49*, 123–157. [CrossRef]
2. Afolalu, O.; Tsoeu, M.S. Enterprise Networking Optimization: A Review of Challenges, Solutions, and Technological Interventions. *Future Internet* **2025**, *17*, 133. [CrossRef]
3. Darwish, T.; Alhaj, T.A.; Elhaj, F.A. Controller placement in software defined emerging networks: A review and future directions. *Telecommun. Syst.* **2025**, *88*, 18. [CrossRef]
4. Kazi, B.U.; Islam, M.K.; Siddiqui, M.M.H.; Jaseemuddin, M. A Survey on Software Defined Network-Enabled Edge Cloud Networks: Challenges and Future Research Directions. *Network* **2025**, *5*, 16. [CrossRef]
5. Ateya, A.A.; Muthanna, A.; Koucheryavy, A. 5G framework based on multi-level edge computing with D2D enabled communication. In Proceedings of the 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon, Republic of Korea, 11–14 February 2018; pp. 507–512.
6. Nunes, B.A.A.; Mendonca, M.; Nguyen, X.N.; Obraczka, K.; Turletti, T. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Commun. Surv. Tutor.* **2018**, *16*, 1617–1634. [CrossRef]
7. Hu, T.; Yi, P.; Zhang, J.; Lan, J. Reliable and load balance-aware multi-controller deployment in SDN. *China Commun.* **2018**, *15*, 184–198. [CrossRef]
8. Lu, J.; Zhang, Z.; Hu, T.; Yi, P.; Lan, J. A Survey of Controller Placement Problem in Software-Defined Networking. *IEEE Access* **2019**, *7*, 24290–24307. [CrossRef]
9. Hock, D.; Hartmann, M.; Gebert, S.; Jarschel, M.; Zinner, T.; Tran-Gia, P. Pareto-optimal resilient controller placement in SDN-based core networks. In Proceedings of the 2013 25th International Teletraffic Congress (ITC), Shanghai, China, 10–12 September 2013; pp. 1–9. [CrossRef]
10. Xu, Y.; He, S.; Zhou, Z.; Xu, J. Redundant Path Optimization in Smart Ship Software-Defined Networking and Time-Sensitive Networking Networks: An Improved Double-Dueling-Deep-Q-Networks-Based Approach. *J. Mar. Sci. Eng.* **2024**, *12*, 2214. [CrossRef]
11. Chen, L.; Lingys, J.; Chen, K.; Liu, F. Auto: Scaling deep reinforcement learning for datacenter-scale automatic traffic optimization. In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18), Budapest, Hungary, 20–25 August 2018; pp. 191–205. [CrossRef]
12. Kumari Rajoriya, M.; Gupta, C.P. SO-CPP: Sailfish optimization-based controller placement in IoT-enabled software-defined wireless sensor networks. *Int. J. Commun. Syst.* **2024**, *37*, e5757. [CrossRef]
13. Asadollahi, S.; Goswami, B.; Sameer, M. Ryu controller's scalability experiment on software defined networks. In Proceedings of the 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), Bangalore, India, 1–2 February 2018; pp. 1–5. [CrossRef]
14. Zhong, Q.; Wang, Y.; Li, W.; Qiu, X. A min-cover based controller placement approach to build reliable control network in SDN. In Proceedings of the NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; pp. 481–487. [CrossRef]
15. Papasani, A.; Varma, G.S.; Prasad Reddy, P.V.; Yannam, V.R. Enhanced capacitated next controller placement in software-defined network with modified capacity constraint. *Int. J. Commun. Syst.* **2025**, *38*, e5979. [CrossRef]
16. Chaudhary, R.; Aujla, G.S.; Kumar, N.; Chouhan, P.K. A comprehensive survey on software-defined networking for smart communities. *Int. J. Commun. Syst.* **2025**, *38*, e5296. [CrossRef]
17. Ospina Cifuentes, B.J.; Suárez, Á.; García Pineda, V.; Alvarado Jaimes, R.; Montoya Benitez, A.O.; Grajales Bustamante, J.D. Analysis of the Use of Artificial Intelligence in Software-Defined Intelligent Networks: A Survey. *Technologies* **2024**, *12*, 99. [CrossRef]
18. Memon, S.A.; Andriukaitis, D.; Markevicius, V.; Navikas, D.; Valinevicius, A.; Zilys, M.; Ramanauskas, R.; Sotner, R.; Jerabek, J.; Klimenta, D. A Survey on Controller Placement Algorithms for IoT Networks in Smart City Environments. In Proceedings of the 2025 IEEE 12th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 15–17 May 2025; pp. 1–6. [CrossRef]
19. Saeedi Goraghani, M.; Afzali, M.; Sharifi, F. A Reliable and Load Balancing Controller Placement Method in Software-Defined Networks. *Int. J. Commun. Syst.* **2025**, *38*, e6059. [CrossRef]
20. Bari, M.F.; Roy, A.R.; Chowdhury, S.R.; Zhang, Q.; Zhani, M.F.; Ahmed, R.; Boutaba, R. Dynamic Controller Provisioning in Software Defined Networks. In Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013), Zurich, Switzerland, 14–18 October 2013; pp. 18–25. [CrossRef]
21. Sahoo, K.S.; Puthal, D.; Obaidat, M.S.; Sarkar, A.; Mishra, S.K.; Sahoo, B. On the placement of controllers in software-defined-WAN using meta-heuristic approach. *J. Syst. Softw.* **2018**, *145*, 180–194. [CrossRef]

22. Ahmadi, V.; Khorramizadeh, M. An adaptive heuristic for multi-objective controller placement in software-defined networks. *Comput. Electr. Eng.* **2018**, *66*, 204–228. [CrossRef]
23. Firouz, N.; Masdari, M.; Sangar, A.B.; Majidzadeh, K. A Hybrid Multi-objective Algorithm for Imbalanced Controller Placement in Software-Defined Networks. *J. Netw. Syst. Manag.* **2022**, *30*, 51. [CrossRef]
24. Salam, R.; Bhattacharya, A. Efficient greedy heuristic approach for fault-tolerant distributed controller placement in scalable SDN architecture. *Clust. Comput.* **2022**, *25*, 4543–4572. [CrossRef]
25. Adekoya, O.; Aneiba, A. An Adapted Nondominated Sorting Genetic Algorithm III (NSGA-III) With Repair-Based Operator for Solving Controller Placement Problem in Software-Defined Wide Area Networks. *IEEE Open J. Commun. Soc.* **2022**, *3*, 888–901. [CrossRef]
26. Tan, L.; Su, W.; Gao, S.; Miao, J.; Cheng, Y.; Cheng, P. Path-flow matching: Two-sided matching and multiobjective evolutionary algorithm for traffic scheduling in cloud data* center network. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3809. [CrossRef]
27. Hou, J.; Tao, T.; Lu, H.; Nayak, A. Intelligent Caching with Graph Neural Network-Based Deep Reinforcement Learning on SDN-Based ICN. *Future Internet* **2023**, *15*, 251. [CrossRef]
28. Troia, S.; Sapienza, F.; Varé, L.; Maier, G. On Deep Reinforcement Learning for Traffic Engineering in SD-WAN. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2198–2212. [CrossRef]
29. Gasior, D. Self-optimizing SD-WAN. In *Artificial Intelligence and Machine Learning*; Soliman, K.S., Ed.; IBIMA-AI 2024. Communications in Computer and Information Science; Springer: Cham, Switzerland, 2025; Volume 2299. [CrossRef]
30. Fu, C.; Wang, B.; Liu, H.; Wang, W. Software-Defined Virtual Private Network for SD-WAN. *Electronics* **2024**, *13*, 2674. [CrossRef]
31. Sminesh, C.N.; Kanaga, E.G.; Roy, A. Optimal multi-controller placement strategy in SD-WAN using modified density peak clustering. *IET Commun.* **2019**, *13*, 3509–3518. [CrossRef]
32. Zafar, A.; Samad, F.; Syed, H.J.; Ibrahim, A.O.; Alohaly, M.; Elsadig, M. An Advanced Strategy for Addressing Heterogeneity in SDN-IoT Networks for Ensuring QoS. *Appl. Sci.* **2023**, *13*, 7856. [CrossRef]
33. Thalapala, V.S.; Guravaiah, K. Fcmcp: Fuzzy c-means for controller placement in software defined networking. *Procedia Comput. Sci.* **2022**, *201*, 109–116. [CrossRef]
34. Sharma, A.; Tokekar, S.; Varma, S. Adaptive Load Balancing Scheme for Software-Defined Networks Using Fuzzy Logic Based Dynamic Clustering. In *Sustainable Communication Networks and Application*; Karrupusamy, P., Balas, V.E., Shi, Y., Eds.; Lecture Notes on Data Engineering and Communications Technologies; Springer: Singapore, 2022; Volume 93. [CrossRef]
35. Jain, A.K.; Kumari, P.; Dhull, R.; Jindal, K.; Raza, S. Enhancing Software-Defined Networking With Dynamic Load Balancing and Fault Tolerance Using a Q-Learning Approach. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e8298. [CrossRef]
36. Abdi Seyedkolaei, A.; Hosseini Seno, S.A.; Moradi, A. Dynamic controller placement in software-defined networks for reducing costs and improving survivability. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4152. [CrossRef]
37. Xu, H.; Chai, X.; Liu, H. A Multi-Controller Placement Strategy for Hierarchical Management of Software-Defined Networking. *Symmetry* **2023**, *15*, 1520. [CrossRef]
38. Ramya, G.; Manoharan, R. Traffic-aware dynamic controller placement in SDN using NFV. *J. Supercomput.* **2023**, *79*, 2082–2107. [CrossRef]
39. Nicol, D.M.; Kumar, R. SDN Resiliency to Controller Failure in Mobile Contexts. In Proceedings of the 2019 Winter Simulation Conference (WSC), National Harbor, MD, USA, 8–11 December 2019; pp. 2831–2842. [CrossRef]
40. Alenazi, M.J.; Cetinkaya, E.K. Resilient placement of SDN controllers exploiting disjoint paths. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3725. [CrossRef]
41. Isong, B.; Molose, R.R.S.; Abu-Mahfouz, A.M.; Dladlu, N. Comprehensive Review of SDN Controller Placement Strategies. *IEEE Access* **2020**, *8*, 170070–170092. [CrossRef]
42. Choumas, K.; Giatsios, D.; Flegkas, P.; Korakis, T. SDN Controller Placement and Switch Assignment for Low Power IoT. *Electronics* **2020**, *9*, 325. [CrossRef]
43. Ge, R.; Liang, H.; Gong, Z.; Hu, C.; Zhou, X.; Cheng, D. Streamlining Data Transfer in Collaborative SLAM Through Bandwidth-Aware Map Distillation. *IEEE Trans. Mob. Comput.* **2025**, *24*, 7554–7567. [CrossRef]
44. Kumar, V.; Patel, R. Security-aware Controller Placement in Software-Defined Networks: Minimizing Attack Surfaces While Maintaining Performance. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 1234–1248.
45. Abdulghani, A.M.; Abdullah, A.; Rahiman, A.R.; Hamid, N.A.; Akram, B.O. Enhancing Healthcare Network Effectiveness Through SD-WAN Innovations. In *Tech Fusion in Business and Society*; Hamdan, R.K., Ed.; Springer: Cham, Switzerland, 2025; pp. 117–130.
46. Frdiesa, M. A Controller Placement Algorithm Using Ant Colony Optimization in Software-Defined Network. *Int. J. Wirel. Inf. Netw.* **2024**, *31*, 142–154. [CrossRef]
47. Yusuf, N.M.; Bakar, K.A.; Isyaku, B.; Abdelmaboud, A.; Nagmeldin, W. Controller placement with critical switch aware in software-defined network (CPCSA). *PeerJ Comput. Sci.* **2023**, *9*, e1698. [CrossRef]

48. Li, C.; Liu, J.; Ma, N.; Zhang, Q.; Zhong, Z.; Jiang, L.; Jia, G. Deep reinforcement learning based controller placement and optimal edge selection in SDN-based multi-access edge computing environments. *J. Parallel Distrib. Comput.* **2024**, *193*, 104948. [CrossRef]
49. Mokhtar, H.; Di, X. Multiple-level threshold load balancing in distributed SDN controllers. *Comput. Netw.* **2021**, *199*, 108510. [CrossRef]
50. Scripts MT: Calculate Distance, Bearing and More Between Latitude. Longitude Points, [cit. 14.9.2012] dostupn na, 2013. Available online: http://www.movable-type.co.uk/scripts/latlong.html (accessed on 22 April 2025).
51. Knight, S.; Nguyen, H.; Falkner, N.; Roughan, M. Realistic network topology construction and emulation from multiple data sources. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 1765–1775. [CrossRef]
52. Guo, Z.; Su, M.; Xu, Y.; Duan, Z.; Wang, L.; Hui, S.; Chao, H.J. Improving the performance of load balancing in software-defined networks through load variance-based synchronization. *Comput. Netw.* **2014**, *68*, 95–109. [CrossRef]
53. Zhong, H.; Lin, Q.; Cui, J.; Shi, R.; Liu, L. An efficient SDN load balancing scheme based on variance analysis for massive mobile users. *Mob. Inf. Syst.* **2015**, *2015*, 241732. [CrossRef]
54. Akram, B.O.; Noordin, N.K.; Hashim, F.; Rasid, M.A.; Salman, M.I.; Abdulghani, A.M. Enhancing reliability of time-triggered traffic in joint scheduling and routing optimization within time-sensitive networks. *IEEE Access* **2024**, *12*, 78379–78396. [CrossRef]
55. Akram, B.O.; Noordin, N.K.; Hashim, F.; Rasid, M.F.; Salman, M.I.; Abdulghani, A.M. Joint scheduling and routing optimization for deterministic hybrid traffic in time-sensitive networks using constraint programming. *IEEE Access* **2023**, *11*, 142764–142779. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

EnvMat: A Network for Simultaneous Generation of PBR Maps and Environment Maps from a Single Image

SeongYeon Oh, Moonryul Jung * and Taehoon Kim *

Graduate School of Metaverse, Sogang University, Seoul 04107, Republic of Korea; dhdev95@gmail.com
* Correspondence: moon@sogang.ac.kr (M.J.); taehoonkim@sogang.ac.kr (T.K.)

Abstract: Generative neural networks have expanded from text and image generation to creating realistic 3D graphics, which are critical for immersive virtual environments. Physically Based Rendering (PBR)—crucial for realistic 3D graphics—depends on PBR maps, environment (env) maps for lighting, and camera viewpoints. Current research mainly generates PBR maps separately, often using fixed env maps and camera poses. This limitation reduces visual consistency and immersion in 3D spaces. Addressing this, we propose EnvMat, a diffusion-based model that simultaneously generates PBR and env maps. EnvMat uses two Variational Autoencoders (VAEs) for map reconstruction and a Latent Diffusion UNet. Experimental results show that EnvMat surpasses the existing methods in preserving visual accuracy, as validated through metrics like L-PIPS, MS-SSIM, and CIEDE2000.

Keywords: generative artificial intelligence; physically based rendering (PBR); environment maps; variational autoencoders (VAEs); latent diffusion models; metaverse; 3D graphics

1. Introduction

Recently, generative artificial neural networks—such as ChatGPT, Stable Diffusion, and Midjourney—have garnered significant attention due to their potential in various creative and practical applications. These generative models have evolved beyond text and image creation to include the generation of realistic 3D graphics, a development expected to play a pivotal role in building immersive metaverse environments.

In academia, research leveraging deep learning technologies for generating 3D graphics is rapidly advancing. Particularly, convolutional neural networks (CNNs) [1] and GAN-based models [2] have been explored to transform low-resolution textures or images into high-resolution outputs. More recently, diffusion-based generative models [3], which address certain limitations of GAN models, are being extensively studied for 3D graphic generation.

To create realistic and coherent 3D graphics, Physically Based Rendering (PBR) is essential. PBR is a rendering technique that simulates the physical interactions between light and materials, enabling computers to realistically portray object appearances. This process involves complex calculations that account for the physical properties of materials, the colors of light sources, and observer viewpoints, necessitating accurate PBR maps, environment maps (env maps), and camera poses.

Traditionally, generating PBR maps has been labor-intensive, requiring designers to meticulously adjust material attributes such as reflectivity, roughness, and metallicity based on real-world observations. Similarly, env maps have typically been created by capturing 360-degree images and applying complex post-processing, imposing significant time and cost constraints.

Applying generative artificial intelligence to these tasks could significantly reduce resource demands, enhancing 3D content quality and production efficiency.

This study aims to propose a diffusion-based neural network model capable of simultaneously generating PBR maps and env maps, which are essential components for realistic 3D rendering.

Developing such a model can substantially improve the efficiency of 3D content production, enabling the creation of more sophisticated and realistic virtual environments, thereby accelerating the advent of the metaverse era. This advancement promises new creative opportunities for 3D artists and developers, as well as more immersive experiences for users.

Although recent studies frequently generate PBR maps from textual or single-image inputs, env maps and camera positions must also be comprehensively integrated to achieve consistent visual outcomes. Without sufficiently incorporating these factors, generated PBR maps risk being constrained to specific conditions, resulting in reduced visual consistency.

For example, basecolor maps (Figure 1) rendered with various environment maps, as shown in Figure 2, can produce significantly different visual results even at the same camera position. Using such inconsistent renderings as inputs to traditional PBR-only generation models further exacerbates inconsistencies, distorting original material properties. This issue specifically affects color fidelity, significantly impacting user experience and immersion.

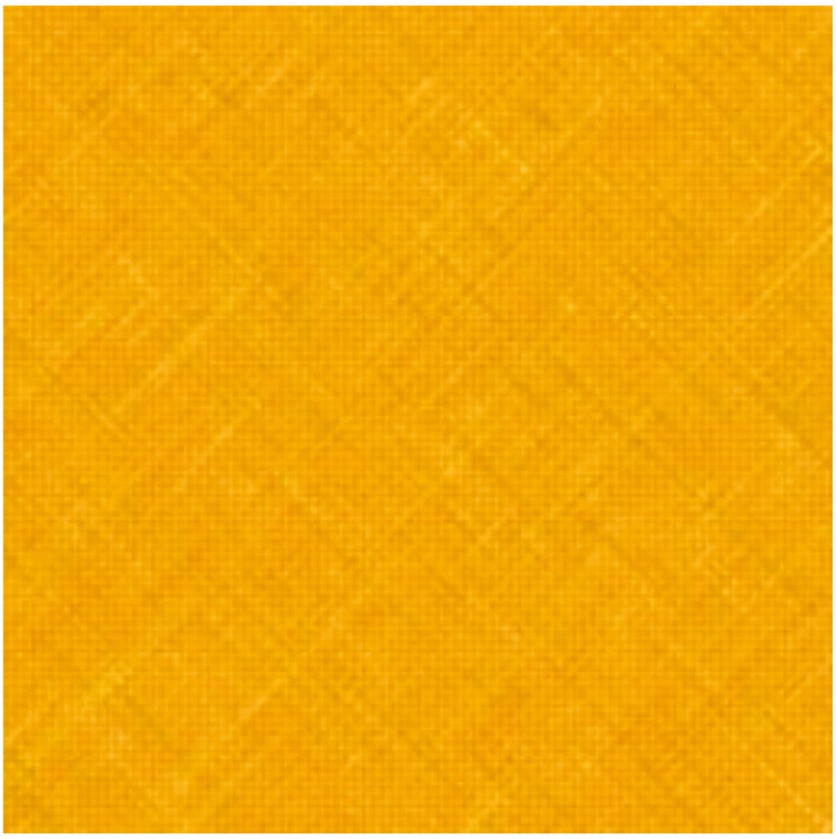


Figure 1. A basecolor map, a type of PBR map containing color information.

To address these issues, this study proposes a neural network model that generates both PBR and env maps simultaneously, more accurately reflecting the original material characteristics. This approach ensures greater visual consistency and realism in generated 3D graphics, ultimately contributing to higher-quality virtual environments.

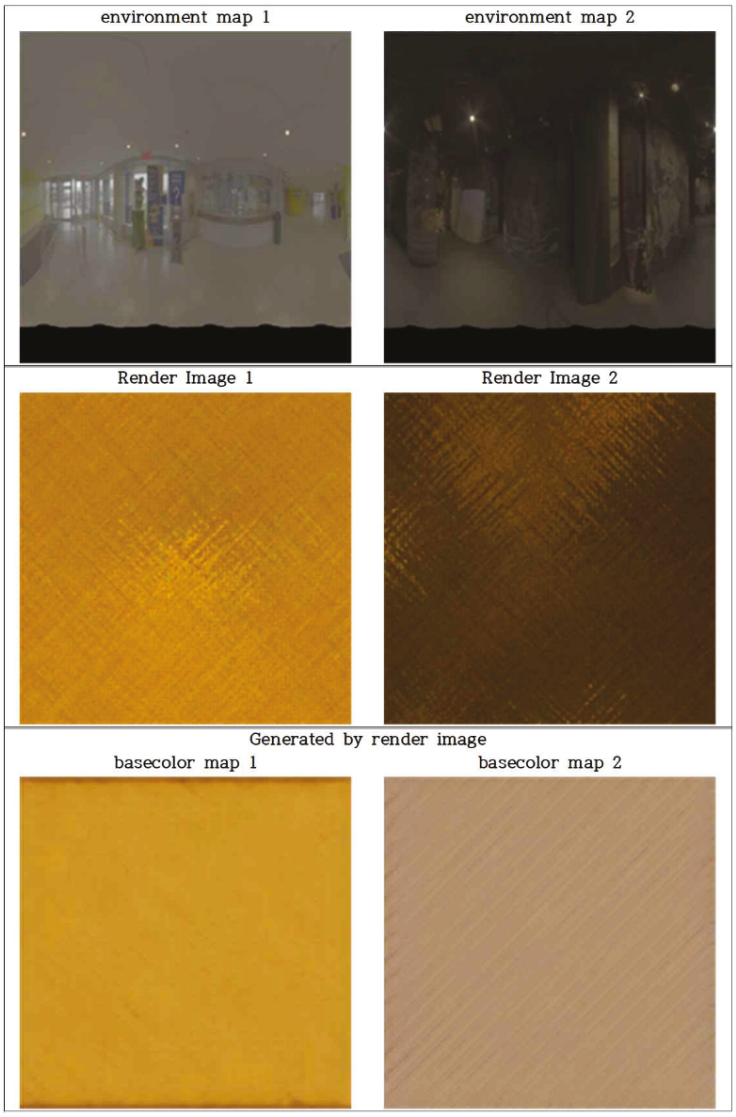


Figure 2. Results generated using images rendered with different environment maps as inputs. The result of rendering the basecolor map in Figure 1 with two different environment maps (1st row) is the Render Image (2nd row); below that are the basecolor maps generated by passing each Render Image as input through the same existing neural network (MatFuse [4]: a neural network that generates PBR map from a single image) (3rd row).

2. Related Work

In this section, we briefly review the previous studies relevant to this research, from foundational material capture principles to recent deep learning-based generative models. We then highlight the key limitations of the existing methods and introduce how our approach differs.

2.1. Material Capture and Estimation

Material capture, the process of reconstructing virtual representations of existing materials, is a fundamentally challenging task. To achieve a faithful reconstruction, one must simultaneously infer intrinsic material properties (typically as Physically Based Rendering, or PBR, maps), environmental lighting (env maps), and camera positions from a limited number of images. These factors are interconnected, creating inherent ambiguities. For instance, even identical materials can appear dramatically different under varying lighting conditions or from different camera viewpoints, making it difficult to isolate their true intrinsic characteristics from image data alone.

2.2. Deep Learning-Based Material Estimation

To address these ambiguities, deep learning has significantly advanced material estimation. Early influential methods—many employing U-Net architectures—have laid the groundwork for the field. For example, the approach described by Deschaintre et al. [5] inspired subsequent Generative Adversarial Network (GAN)-based methods like SurfaceNet [6]. Another notable model, MaterialGAN [7], utilized a StyleGAN2-based architecture [2] to construct a latent space optimized for reproducing input images, thereby generating realistic BRDF maps.

2.3. Generative Models for Material Synthesis

More recently, the landscape of generative modeling has been reshaped by the success of Diffusion Models (DMs) [8], which have begun to replace GANs due to their stable training and their ability to generate diverse, high-quality images. To mitigate their significant computational cost, Latent Diffusion Models (LDMs) like Stable Diffusion [3] were introduced. LDMs operate in a compressed latent space created by a pretrained Variational Autoencoder (VAE) [9], drastically reducing computational overhead while preserving quality.

This powerful LDM framework has been successfully applied to material generation. Models like Text2Mat [10] and MatFuse [4] demonstrate the ability to generate PBR maps from a variety of conditional inputs, such as text or images. Further advancements are focused on generating tileable textures, with methods like MaterIA [11] and ControlMat [12] introducing techniques to ensure seamless repetition, enabling fine-grained control over the final output.

2.4. Differences from Existing Work

Despite these advancements, current diffusion-based approaches have predominantly focused on generating PBR maps while operating under highly constrained environmental conditions. This focus can cause serious color consistency degradation issues in the rendered environment. For example, a state-of-the-art model like MatFuse [4] was trained to infer PBR maps by capturing materials under only five predefined environment maps and from a single, fixed camera position. This reliance on a limited set of conditions represents a clear, fundamental limitation.

The critical drawback of training and inferring within such a restricted environment is the model's failure to preserve the original, intrinsic properties of the material. This issue is particularly pronounced in the color domain, which is highly sensitive to the environment map used during rendering. As seen in Figure 2, rendering and inferring the same material from different environment maps can result in noticeable color distortion, which can result in loss of visual fidelity.

Therefore, this research originates from the hypothesis that these limitations can be overcome by designing a model that simultaneously estimates both the PBR maps and the corresponding environment map. By explicitly modeling the environment as part of the generation process, we aim to disentangle the material's appearance from the lighting conditions, allowing the model to learn a more robust representation of intrinsic material properties. This approach is designed to resolve the visual inconsistencies seen in previous work and significantly improve the realism and consistency of the generated 3D assets.

3. Methodology

3.1. Datasets

All data were organized in a train/validation/test = 8:1:1 ratio, and this section focuses on the amount of data used for training.

3.1.1. PBR Maps

This study utilized the MatSynth dataset [13] provided by Adobe, comprising 4006 materials at 4K resolution, each with various PBR maps. We selected four map types—basecolor, normal, roughness, and metallic. These four maps collectively define our PBR maps.

Following MatSynth's preprocessing method, each material was rotated at 45-degree intervals, creating eight rotations. These images were cropped to non-overlapping sizes (one 4K image, four 2K images per rotation) as shown in Figure 3, and then resized to 256 × 256 pixels.

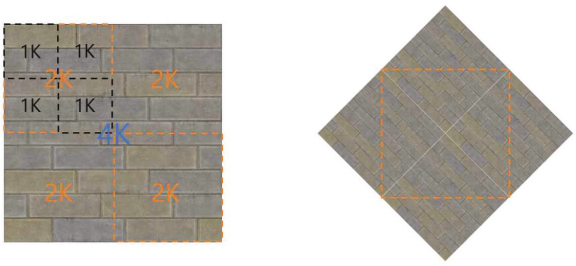


Figure 3. Image cropping method (**left**) and cropping method after rotation (**right**).

To manage dataset size and ensure diversity, we randomly selected 1000 materials across 14 categories (≈40,000 images after augmentation), choosing 72 samples from each of 13 categories and 64 samples from the Blend category.

3.1.2. Environment Maps (Env Maps)

We used the Laval indoor dataset [14], comprising 2300 HDR environment maps initially in EXR format. Following DPI [15], these were converted to PNG images at a resolution of 256 × 256 pixels using Global Invertible Mapping.

To further augment the dataset, each env map was horizontally shifted in 20-pixel increments (Figure 4), generating ten additional images per original map. Additionally, horizontally mirrored versions of each image were similarly shifted (Figure 5), producing 11 more images. This resulted in a total of approximately 50,600 augmented images, from which 40,000 were selected for training.

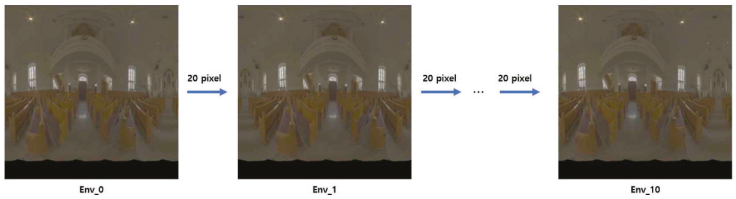


Figure 4. Data augmentation using rolling method.

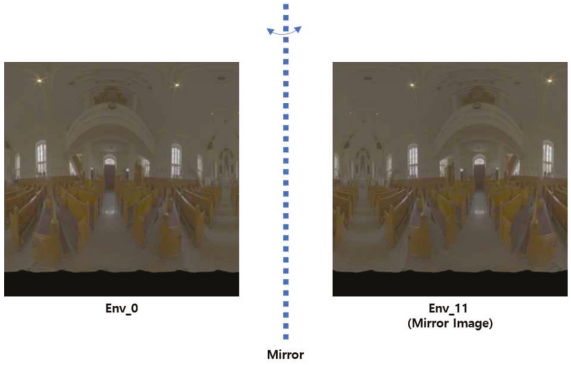


Figure 5. Data augmentation using horizontal mirroring.

3.1.3. Rendered Images

Rendered images serve as conditions (y) for training the Latent Diffusion Model (LDM). Rendering was performed using the Mitsuba 3 Renderer. Each rendered image required one PBR map and one env map. However, exhaustively rendering all combinations (potentially 1.6 billion) was impractical due to computational limitations.

Therefore, for each PBR map, we randomly selected eight original (non-augmented) env maps for rendering. Augmented env maps were intentionally avoided to minimize model bias, as augmented versions share similar distributions. This process generated a final dataset of 320,000 pairs of [PBR map, env map, rendered image] used for training.

3.2. Model Training

Overall structure of proposed model is depicted in Figure 6.

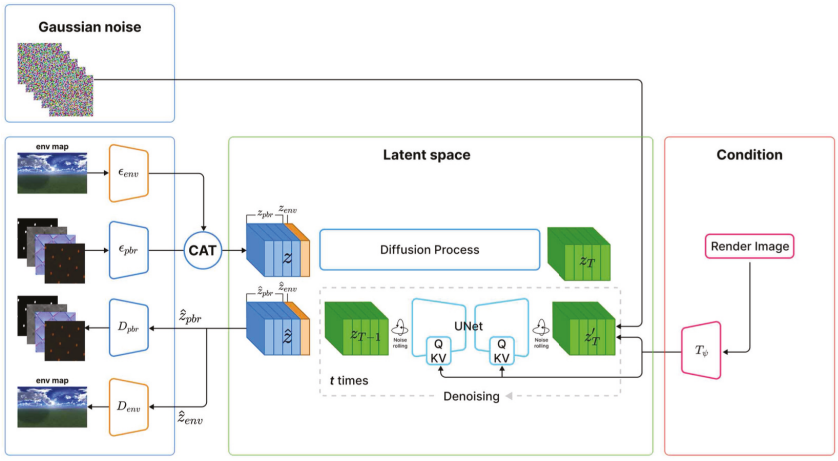


Figure 6. Overall architecture of proposed model. It has two PBR VAEs and env VAEs. In addition, the latent vectors (z_{pbr}: 12 channels, z_{env}: 3 channels) that passed through each encoder are concatenated and used as the input of the Latent U-Net. The rendered image is used as the condition of the U-Net.

3.2.1. PBR VAE

We implemented the PBR VAE using VQ-GAN [9], following a similar approach as MatFuse [4]. Each of the four PBR maps (basecolor, normal, roughness, metallic) was individually encoded into latent vectors of size 32×32. These four latent vectors were concatenated to form a combined latent vector (z_{pbr}). A single unified decoder then reconstructed these latent representations back to their original maps.

The training loss function (L_{total}) combined pixel–space L2 loss, perceptual LPIPS loss, patch-based adversarial loss, codebook commitment loss, and rendering loss.

Training employed the Adam optimizer with mini-batch gradient descent, a batch size of 4, and a learning rate of 10^{-4}. Training continued up to 120,000 iterations, until validation loss convergence.

3.2.2. ENV VAE

The env VAE was also based on VQ-GAN [9], structured with a single encoder–decoder pair. Env maps were compressed into latent representations (z_{env}) of size 32×32 and then reconstructed.

The loss function included pixel–space L2 loss, perceptual LPIPS loss, patch-based adversarial loss, and codebook commitment loss (Excluding rendering loss from the loss function of Section 3.2.1).

Similar training parameters as the PBR VAE were applied—mini-batch gradient descent with Adam optimizer, batch size of 4, learning rate of 10^{-4}, and 120,000 iterations.

The final architecture incorporating both VAEs is depicted in Figure 7.

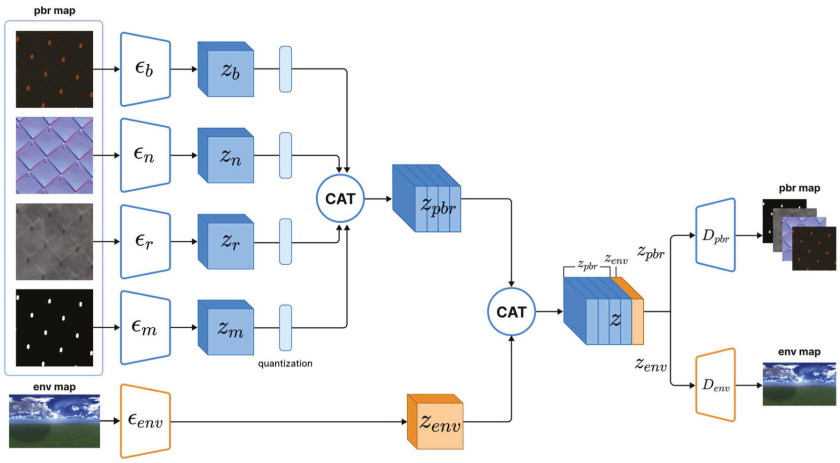


Figure 7. Structure of the two Variational Autoencoders (VAEs). The PBR encoder processes four types of basecolor (b), with normal (n), roughness (r), and metallic (m) as three separate channels, and concatenates them into one latent vector. The env encoder has a structure that encodes the env map into a three-channel latent vector (z_{env}) and concatenates it with z_{pbr}. The decoder performs the above process in reverse.

3.2.3. Latent Diffusion Model (LDM)

Our generative model is based on the Latent Diffusion Model (LDM), which learns to reverse a forward diffusion process in a latent space. The LDM is trained to denoise a noisy latent vector z_t conditioned on an input image y, which, in our case, is the rendered image. This process is guided by a UNet-based noise predictor, ϵ_θ, which is optimized via the following objective function:

$$L_{\text{LDM}} = \mathbb{E}_{z,\epsilon\sim N(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right] \quad (1)$$

In this objective, the model ϵ_θ (a UNet) is trained to predict the original noise ϵ from a noisy latent z_t. The clean latent z_0 is produced by our pre-trained VAE, and the noisy latent z_t is created by adding noise $\epsilon \sim \mathcal{N}(0,1)$ at a specific timestep t. The prediction is guided by the conditioning input y, which is processed into a vector sequence by a dedicated encoder τ_θ and injected into the UNet via cross-attention.

The training utilizes the AdamW optimizer with a batch size of 32 and a learning rate of 10^{-4}, continuing for 100,000 iterations. During inference, a linear schedule is used for β, and noise is removed using the DDIM (Denoising Diffusion Implicit Model) sampling schedule.

4. Experiment

4.1. Evaluation Method

The primary objective of this study is to verify whether simultaneously generating PBR maps and environment maps (env maps) better preserves original PBR map characteristics compared to generating PBR maps alone. Specifically, we focus our evaluation on the basecolor map, as it is most significantly influenced by changes in the env map. To assess the similarity between the generated basecolor maps and the original maps, we employ the following three metrics:

- **L-PIPS** measures perceptual similarity by computing differences between deep feature maps obtained from pretrained vision models such as VGG-16. This metric effectively captures high-level visual similarities including textures, styles, and structures. An L-PIPS score closer to 0 indicates higher similarity between the two images.

- **MS-SSIM** is an extended version of the Structural Similarity Index Measure (SSIM), evaluating structural similarities at multiple scales. While SSIM comprehensively evaluates luminance, contrast, and structural similarity, MS-SSIM enhances this evaluation across various resolutions, providing a more robust measure of structural similarity. An MS-SSIM score closer to one indicates higher similarity.
- **CIEDE2000** measures color difference based on the Lab color coordinates (L: luminance, a: green-to-red axis, b: blue-to-yellow axis). This metric accounts for human visual sensitivity to luminance, chroma, and hue. A higher CIEDE2000 value indicates a greater color difference. Generally, differences below three are barely noticeable or indistinguishable to the human eye, differences between three and six are clearly distinguishable, and differences above six are considered significantly noticeable.

For evaluation, each test PBR map was rendered with all 225 non-augmented test environment maps, generating 225 rendered images per PBR map. This procedure was consistently applied to 65 PBR maps, selecting five maps from each of the 13 test categories, producing a total of 14,625 rendered images. These images were then used as inputs to both our proposed model (EnvMat) and the baseline model (MatFuse) to generate respective PBR maps for comparison. The MatFuse model was trained on the identical dataset and iterations as the EnvMat model for a fair comparison.

4.2. Training Results

4.2.1. Baseline Model Reproduction

Prior to evaluating our proposed model, we first verified whether the baseline model was accurately reproduced. At the start of our research, only the source code of the baseline model was publicly available without any pretrained checkpoints. Additionally, the baseline model was originally trained on a different dataset than the one used in this study. Therefore, reproducing and validating the baseline model under identical training conditions is essential before proceeding with further comparisons.

To assess the quality of the reproduced baseline model, we evaluated the PBR maps (basecolor, roughness, normal, metallic) generated from the rendered images, employing the same quantitative metrics as the original study, namely FID-score and CLIP-IQA, along with qualitative assessments. These evaluations utilized rendered images created with an arbitrarily selected, identical env map to ensure consistency with the original paper.

Quantitatively, we measured FID-score and CLIP-IQA and compared our results (Table 1) with those of the original baseline study. For CLIP-IQA, we adopted the original high-quality and low-quality evaluation criteria. Notably, differences in CLIP-IQA scores between our reproduced model and the original baseline model are mainly attributed to differences in rendered image quality used for evaluation. CLIP-IQA heavily depends on the intrinsic quality of ground truth (GT) renderings. The rendered images employed in our study are of higher quality compared to those in the baseline study, as the MatSynth dataset utilized here is more recent and higher quality than the dataset used previously. Thus, our reproduced baseline model demonstrated CLIP-IQA scores close to the GT, indicating successful reproduction.

However, our reproduced model showed relatively higher differences in the FID-score compared to the original baseline. This difference is primarily due to the original model being trained on a larger dataset with substantially more training iterations. Specifically, our model was trained with 320,000 (PBR-rendered image) pairs for 120,000 (VAE) and 100,000 (LDM) iterations, whereas the baseline study trained with approximately ten times more data, consisting of 3,200,000 pairs for over 4,000,000 (VAE) and 500,000 (LDM) iterations. Given these considerable differences in training data volume and iterations, it is natural that the original baseline model achieved superior FID-scores.

Table 1. Baseline model reproduction. Quantitative evaluation comparing MatFuse (original paper), reproduced MatFuse (baseline), and proposed EnvMat model.

Model	CLIP-IQA (↑)	FID (↓)
GT Render Image	0.838 ± 0.030	-
MatFuse (paper)	0.431 ± 0.151	158.53
MatFuse (baseline)	0.829 ± 0.026	196.32
EnvMat (ours)	0.831 ± 0.039	188.02

The training process for this study was terminated early when the loss function converged. The reason for the lower number of training iterations compared to the MatFuse paper is a natural consequence of our dataset size being approximately one-tenth of theirs. Due to computational resource and time constraints, it was not feasible to preprocess and train on data at the same scale as MatFuse. Consequently, our model's loss converged within fewer iterations, which can be interpreted as an aspect of training efficiency.

Additionally, the quantitative comparative experiments in the Appendix A (Figures A1–A7) confirm that the reproduced MatFuse produces results at a level suitable for the experiments.

Despite these differences, our reproduced baseline model achieved satisfactory results both quantitatively and qualitatively, verifying its suitability for comparative evaluation with our proposed model (EnvMat).

4.2.2. Qualitative Evaluation

The figures in the Appendix A (Figures A8–A21) illustrate comparisons of basecolor maps generated by the proposed EnvMat and the baseline MatFuse model. Each figure includes the same original PBR map (top-left in the first row), rendered images using different env maps (top-right in the first row and second row), and the resulting basecolor maps generated by EnvMat (third row, left) and MatFuse (third row, right).

In the rendered images, significant color distortion due to different environment maps is evident when compared to the original basecolor maps. Observing the generated outputs, EnvMat consistently produced basecolor maps whose colors closely matched the original, while MatFuse exhibited noticeable deviations.

These qualitative results suggest that MatFuse tends to reflect the distorted color information from rendered images rather than the original basecolor map. In contrast, EnvMat effectively mitigates color distortions from heavily altered rendered images, producing results closer to the original basecolor maps.

4.2.3. Quantitative Evaluation

To verify the effectiveness of our proposed model, we performed a quantitative evaluation that also served as an **ablation study** for our primary contribution. We compared our **EnvMat** model against the **MatFuse** architecture under identical training and evaluation conditions. Since the **MatFuse** architecture was equivalent to our **EnvMat** model with the **Environment Map VAE (env VAE)** component removed, this comparison directly isolates and measures the impact of simultaneously generating environment maps.

We quantitatively evaluated our results using the metrics L-PIPS, MS-SSIM, and CIEDE2000. Following the evaluation protocol described in Section 4.1, we calculated the average similarity scores between the 14,625 basecolor maps generated from each model and their corresponding original basecolor maps. Table 2 summarizes the results.

The results in Table 2 clearly validate our approach. Interpreted as an **ablation study**, the significant performance drop observed in the MatFuse model across all metrics, especially the 10-point degradation in the CIEDE2000 color accuracy score, directly demonstrates that the **env VAE is essential for preserving color fidelity**. As a comparison to the baseline

architecture, EnvMat sets a new performance benchmark, confirming that our method of simultaneous generation yields basecolor maps more closely aligned with the original.

Table 2. Quantitative comparison between EnvMat (our model) and MatFuse (baseline).

Model	L-PIPS (↓)	MS-SSIM (↑)	CIEDE2000 (↓)
MatFuse (Baseline)	0.6899	0.4861	47.4045
EnvMat (Proposed)	0.6476	0.5195	37.2581

4.3. Performance Analysis

To evaluate the practical applicability of our proposed model, we analyzed its real-time performance in terms of inference time and memory usage, as requested by the reviewer. We measured the average time required to generate a single output and the peak GPU memory consumption during inference. Both EnvMat and the MatFuse baseline were evaluated on the same NVIDIA A6000 GPU (Nvidia Corporation, Santa Clara, CA, USA). The results are summarized in Table 3.

Table 3. Inference time and memory usage comparison.

Model	Inference Time (s)	Memory Usage (MiB)
MatFuse (Baseline)	3.6–3.7	8232
EnvMat (Proposed)	3.6–3.8	8498

The analysis reveals a compelling finding—despite utilizing an additional VAE for environment maps, EnvMat exhibits a comparable inference time to the baseline. Furthermore, the increase in memory usage is approximately 266 MiB, a negligible overhead of about 3.2%. These results demonstrate that our method achieves a significant improvement in output quality and color fidelity, with almost no additional computational cost. This highlights the efficiency of our approach and its strong potential for practical applications.

5. Conclusions and Future Work

5.1. Conclusions

Based on the experimental results presented in Section 4, we demonstrate that the proposed EnvMat model, which simultaneously generates environment maps and PBR maps, consistently produced outputs closer to the original maps compared to the baseline model, which generates only PBR maps. Our qualitative and quantitative evaluations confirm that EnvMat achieved superior performance in terms of L-PIPS and MS-SSIM metrics. Particularly, EnvMat showed significant improvement in the CIEDE2000 metric, achieving differences greater than six, indicating clearly distinguishable enhancements in color fidelity.

These results confirm that simultaneously predicting env maps alongside PBR maps significantly mitigates color distortions introduced by varying environmental lighting, effectively preserving the original characteristics of the PBR maps.

Furthermore, our performance analysis revealed that these significant quality improvements are achieved with negligible additional computational cost, highlighting the efficiency and practical viability of our approach.

5.2. Limitations

This study acknowledges that the environment map dataset used was limited to indoor scenes, which restricts the model's generalizability across diverse environments. This was primarily due to the practical availability of suitable datasets during the research period.

Furthermore, a significant difference in training data scale exists when comparing our work to the original MatFuse baseline paper. While the MatFuse paper trained on a 3.2M-scale dataset, our experiments were conducted with approximately one-tenth of that data. This was a consequence of physical limitations encountered during data preprocessing; the rendering of corresponding images for each material and environment map proved to be exceptionally time-consuming and computationally intensive. Due to these resource constraints, training on a scale equivalent to MatFuse was not feasible within the scope of the current research.

Our qualitative analysis revealed that the model struggled to accurately reproduce reddish-toned lighting environments (Figures A22 and A23). We attribute this primarily to an imbalance in the training data distribution.

Recognizing this challenge, we reproduced the MatFuse model using an equivalently scaled dataset to our own. This allowed for a direct comparison between the reproduced MatFuse model and our proposed model. The results clearly demonstrated a significant performance enhancement, particularly in color fidelity, as our framework achieved a remarkable improvement of over six in the CIEDE2000 metric.

In real-world physical environments, purely red ambient light conditions are exceptionally rare, especially when compared to common lighting scenarios like sunsets (which are more orange) or general illumination. Consequently, the vast majority of publicly available environment map datasets do not contain a sufficient number of these reddish-toned environments.

As a result, our model, having been trained on a limited amount of reddish-toned data, exhibited poor generalization performance in this color space, tending to produce inaccurate colors or artifacts. To address this, data augmentation strategies involving the intentional rendering of reddish-toned environments would be necessary. While beyond the scope of this paper, this represents a crucial direction for future research to enhance the model's robustness.

5.3. Future Work

To build upon our findings, we propose the following directions for future research:

- **Increased Generalization through Expanded Datasets:** Future research should employ larger and more diverse datasets to further enhance the model's generalization capability. Expanding the training dataset with additional MatSynth materials, env maps featuring diverse color distributions, as well as env maps captured from various environments (including outdoor and studio lighting conditions), will substantially improve the robustness and applicability of the proposed model.
- **Incorporation of Render Loss in UNet Training:** While Render Loss is often beneficial for optimizing physically based rendering (PBR) assets by evaluating the difference between rendered outputs generated from predicted maps and original renderings, its direct application to our Latent UNet architecture presents inherent structural challenges. Render Loss operates by comparing renderings at the pixel level. However, our Latent UNet operates and generates representations within a latent space, not the pixel space, thereby preventing the direct rendering of intermediate PBR and environment maps for Render Loss computation during its training. Although this loss was effectively employed during VAE training, its integration into the UNet training process remains a significant challenge. This characteristic is common in Latent Diffusion Models. Consistent with the previous work utilizing UNets in similar latent spaces, we did not employ Render Loss during UNet training. Future research should explore methods to effectively integrate Render Loss into UNet training, which could lead to even closer alignment with original renderings and potentially further improve the accuracy of generated results.

Author Contributions: Conceptualization, S.O.; methodology, S.O. and M.J.; software, S.O.; validation, S.O. and T.K.; formal analysis, S.O.; investigation, T.K.; resources, M.J.; data curation, S.O.; writing—original draft preparation, S.O.; writing—review and editing, T.K.; visualization, S.O.; supervision, T.K.; project administration, T.K.; funding acquisition, T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by IITP and MSIT of Korea through the Graduate School of Metaverse Convergence Program (RS-2022-00156318), and by MCST of Korea through the KOCCA grant (RS-2023-00219237) as part of the Culture, Sports, and Tourism R&D Program.

Data Availability Statement: The original data presented in the study are openly available in MatSynth at https://huggingface.co/datasets/gvecchio/MatSynth.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

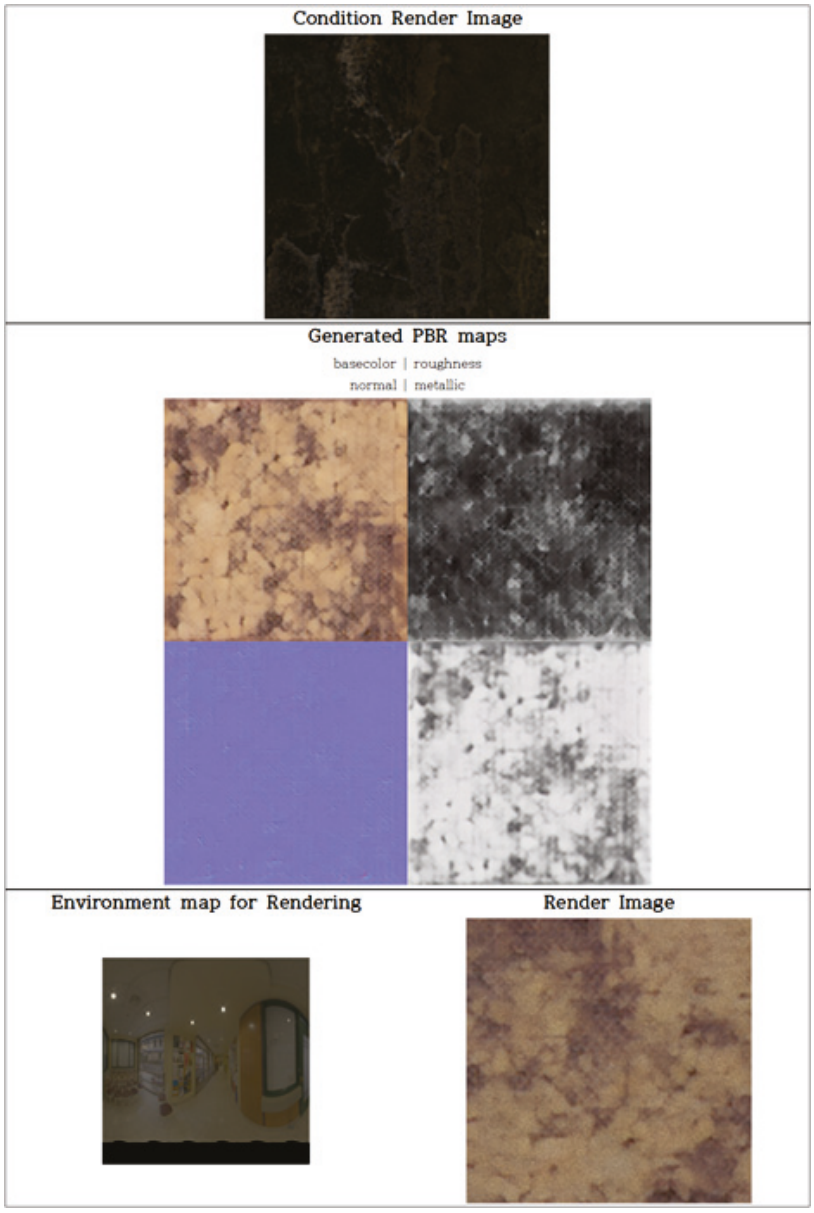


Figure A1. Reproduced MatFuse results. The PBR map (row 2) generated via Condition Render Image (row 1) and the Render Image (row 3, column 2) rendered with a randomly selected env map (row 3, column 1).

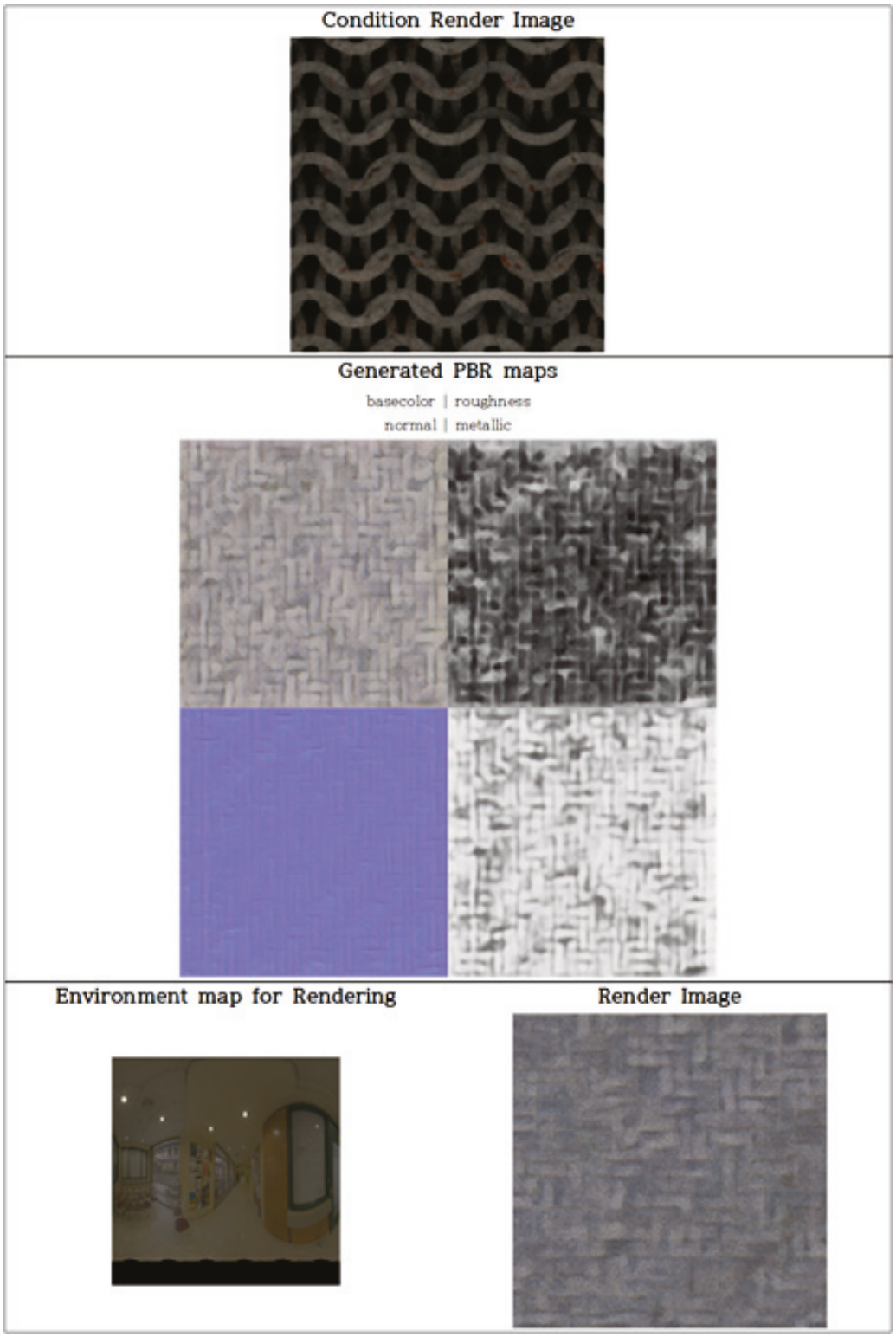


Figure A2. Reproduced MatFuse results. The PBR map (row 2) generated via Condition Render Image (row 1) and the Render Image (row 3, column 2) rendered with a randomly selected env map (row 3, column 1).

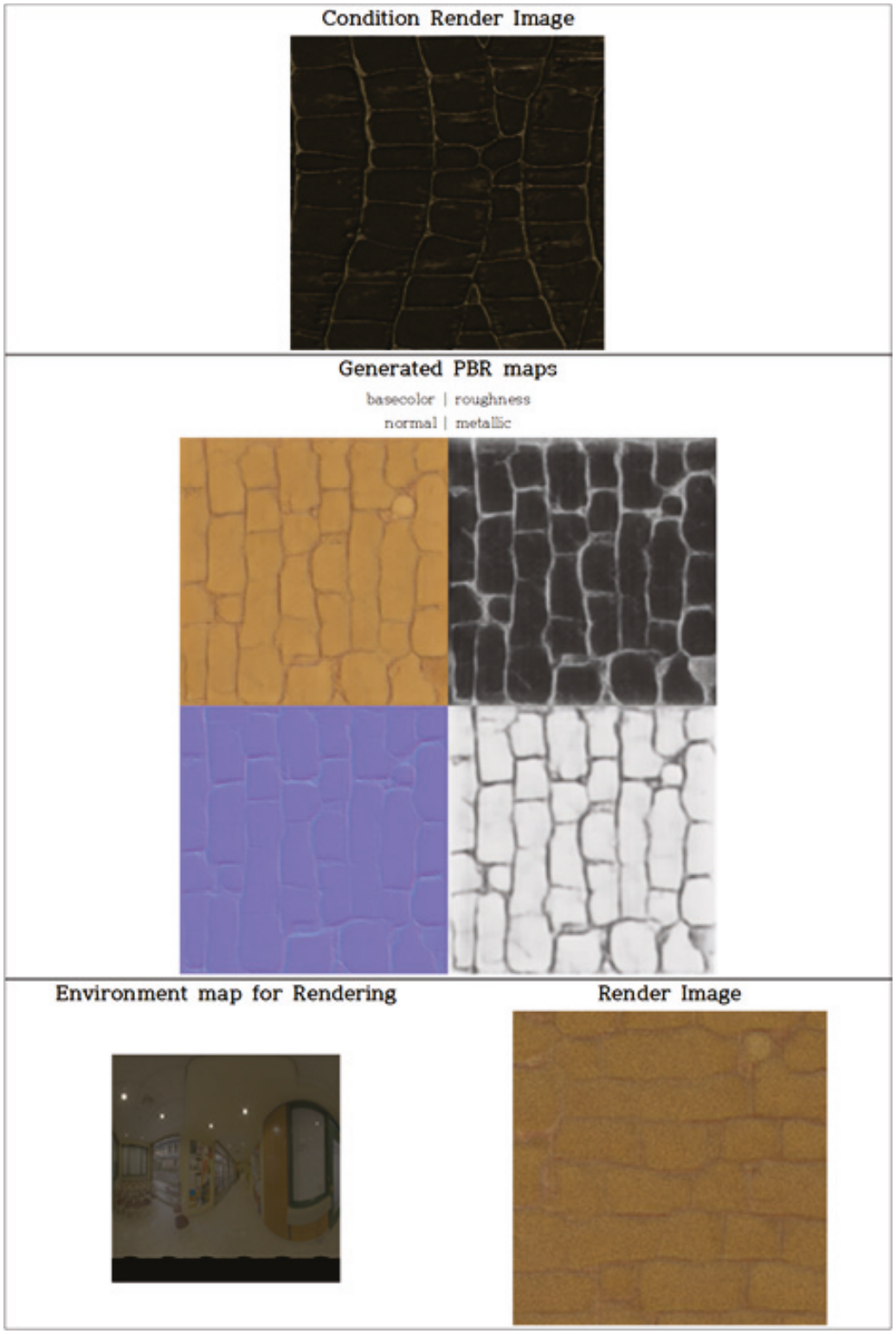


Figure A3. Reproduced MatFuse results. The PBR map (row 2) generated via Condition Render Image (row 1) and the Render Image (row 3, column 2) rendered with a randomly selected env map (row 3, column 1).

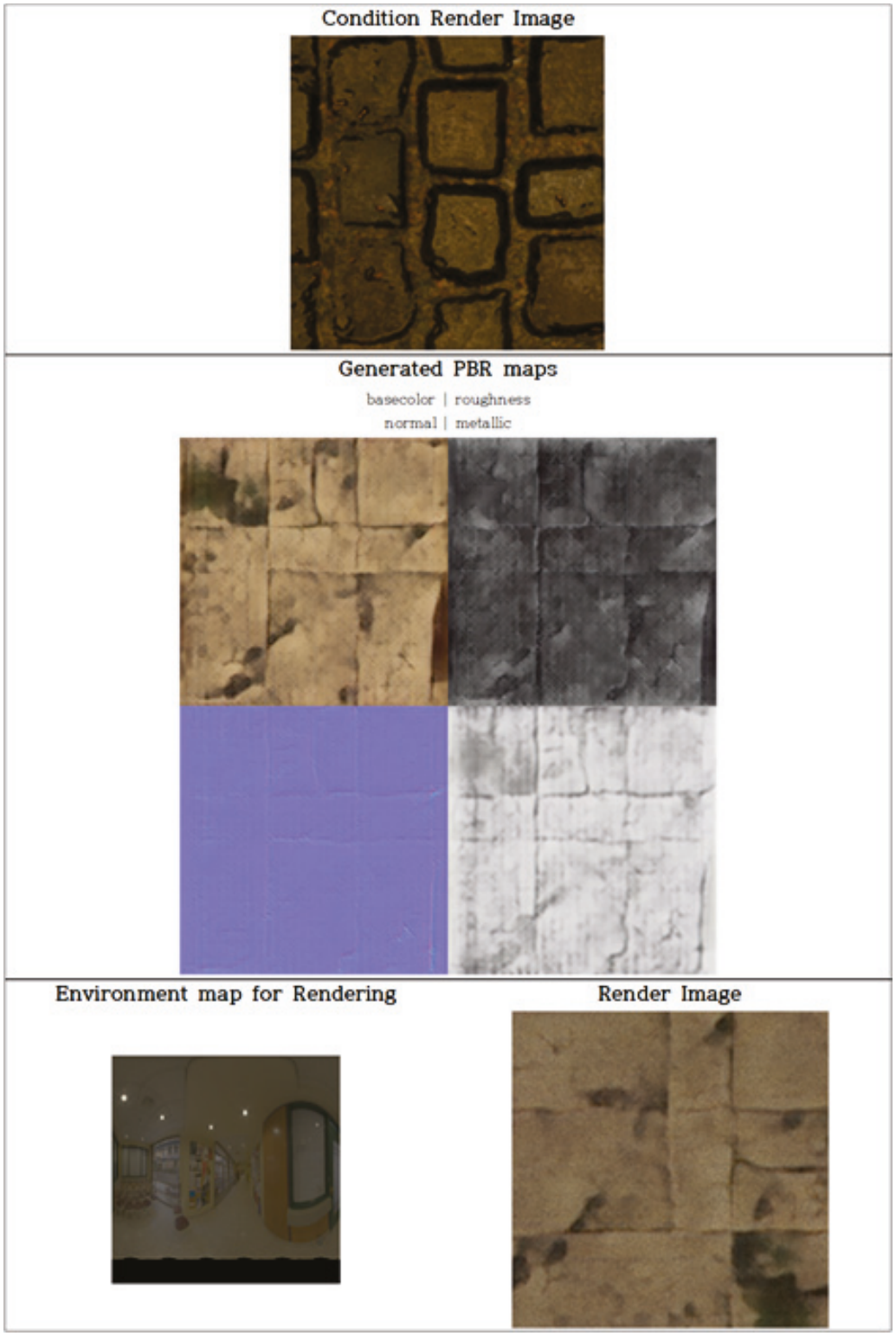


Figure A4. Reproduced MatFuse results. The PBR map (row 2) generated via Condition Render Image (row 1) and the Render Image (row 3, column 2) rendered with a randomly selected env map (row 3, column 1).

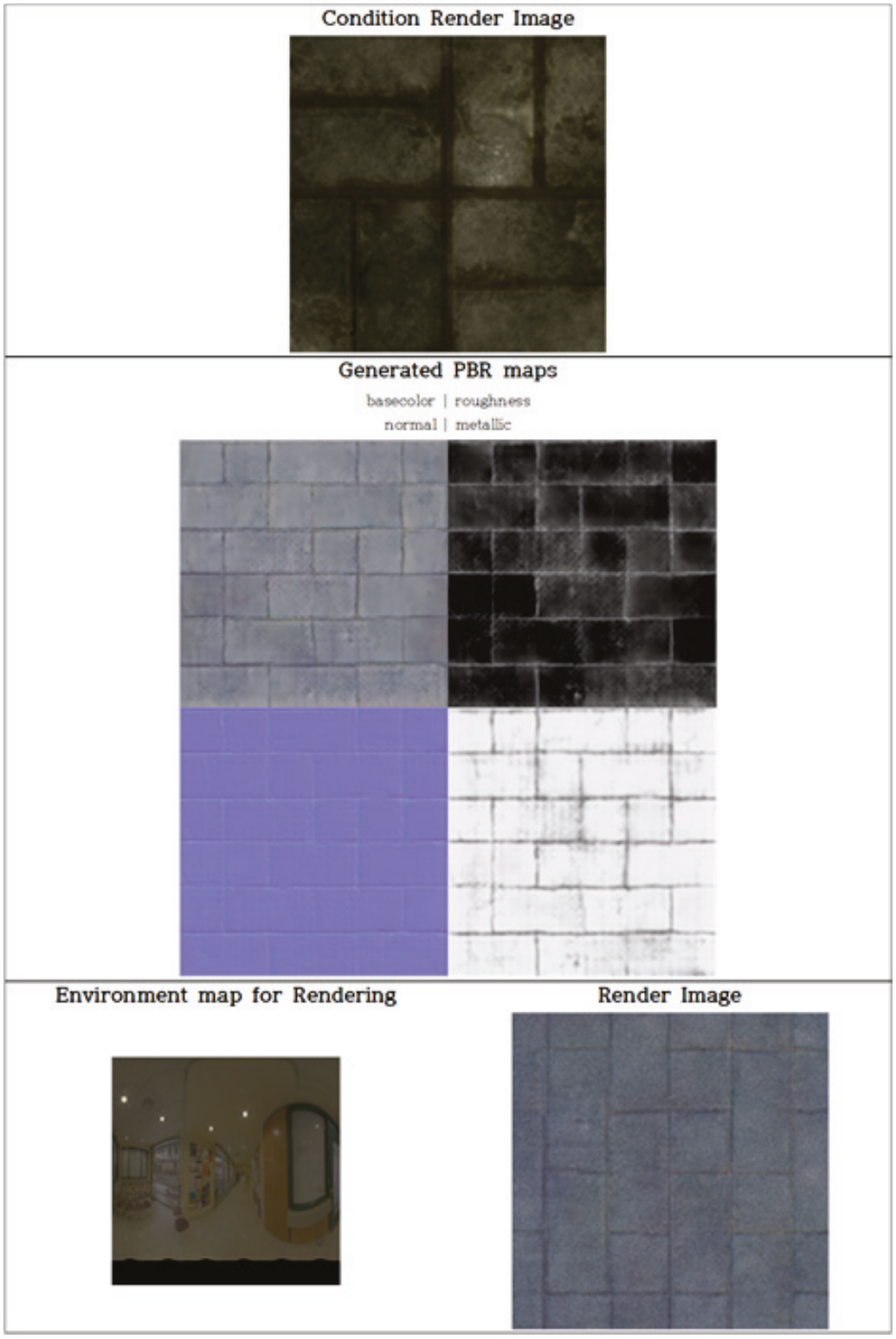


Figure A5. Reproduced MatFuse results. The PBR map (row 2) generated via Condition Render Image (row 1) and the Render Image (row 3, column 2) rendered with a randomly selected env map (row 3, column 1).

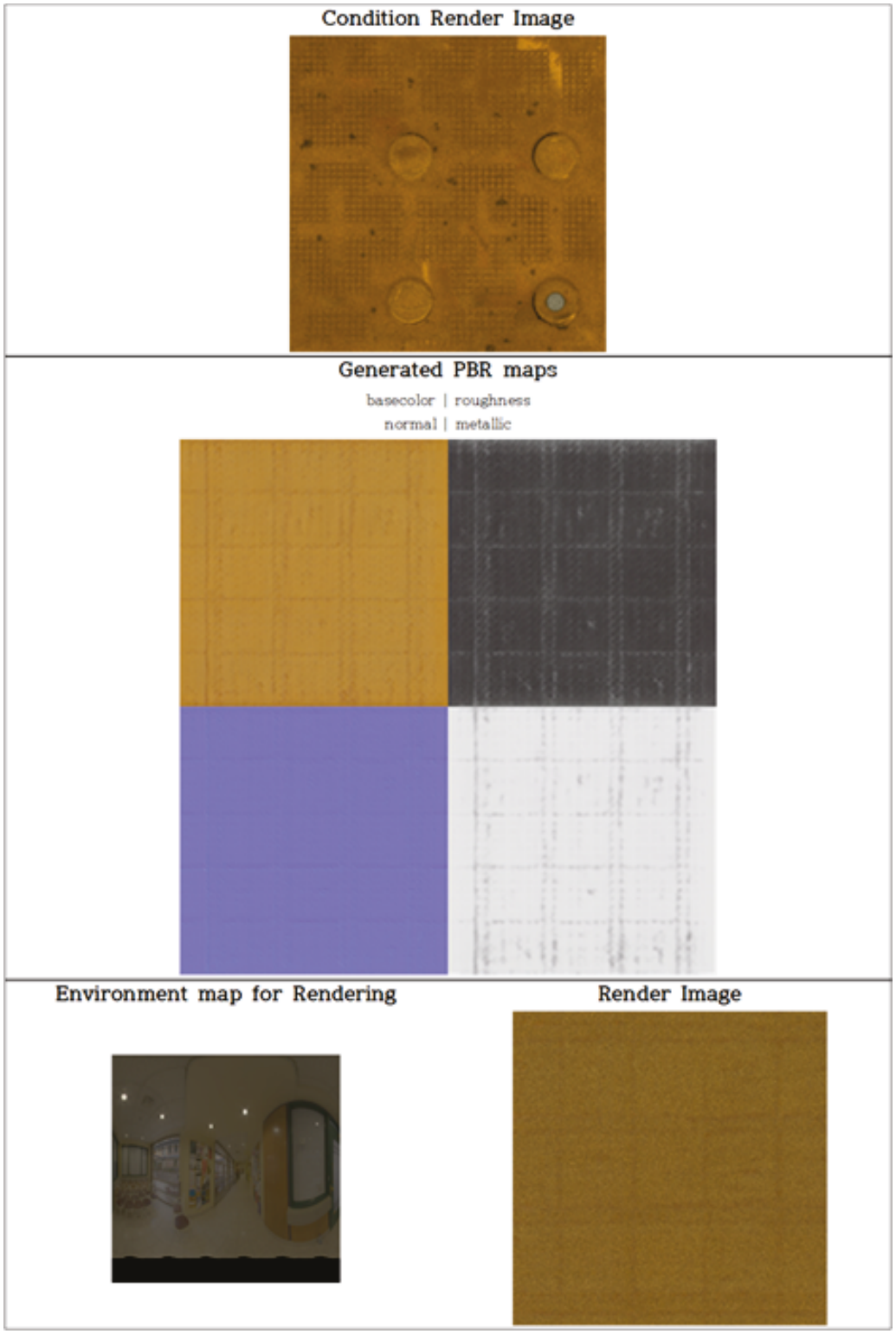


Figure A6. Reproduced MatFuse results. The PBR map (row 2) generated via Condition Render Image (row 1) and the Render Image (row 3, column 2) rendered with a randomly selected env map (row 3, column 1).

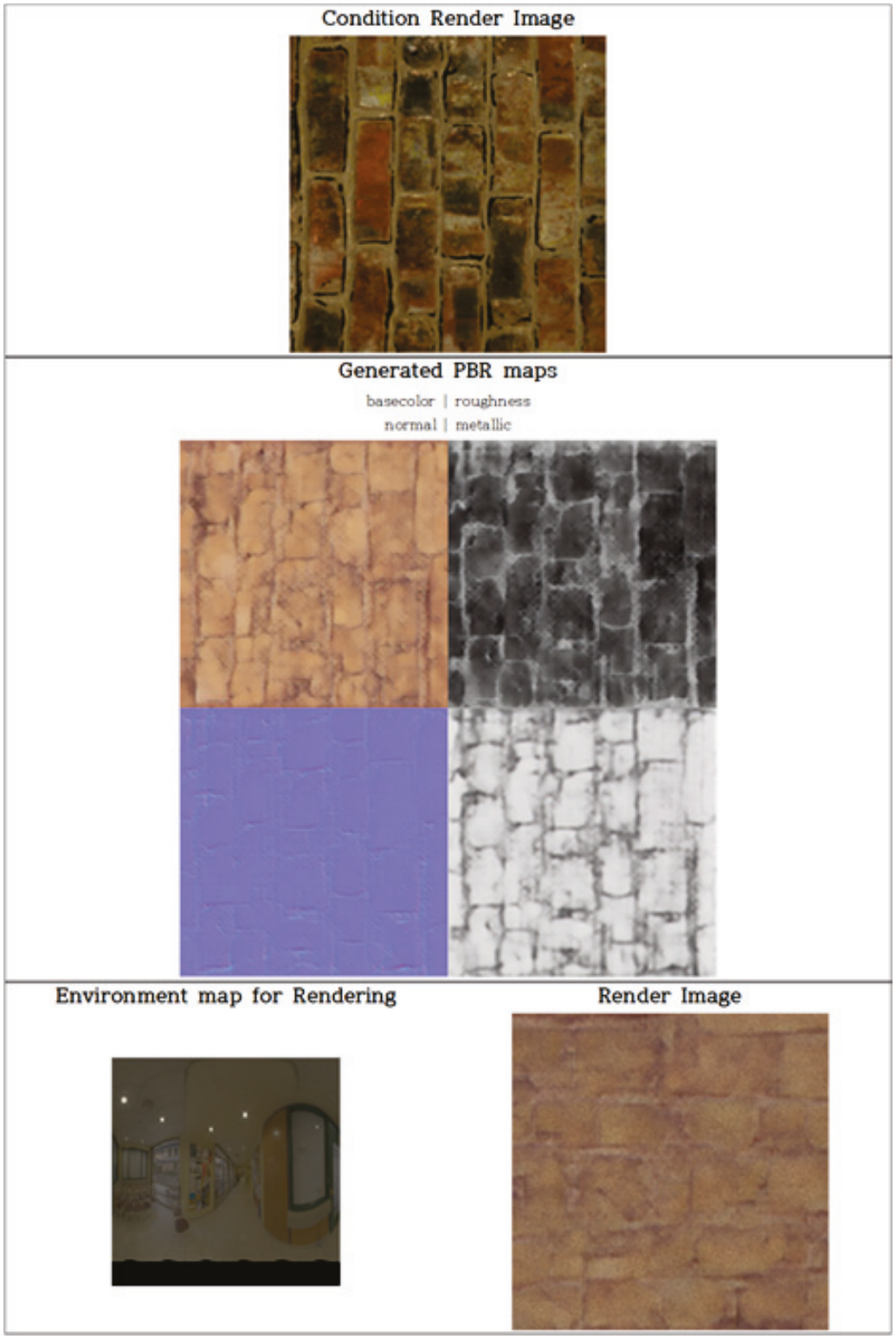


Figure A7. Reproduced MatFuse results. The PBR map (row 2) generated via Condition Render Image (row 1) and the Render Image (row 3, column 2) rendered with a randomly selected env map (row 3, column 1).

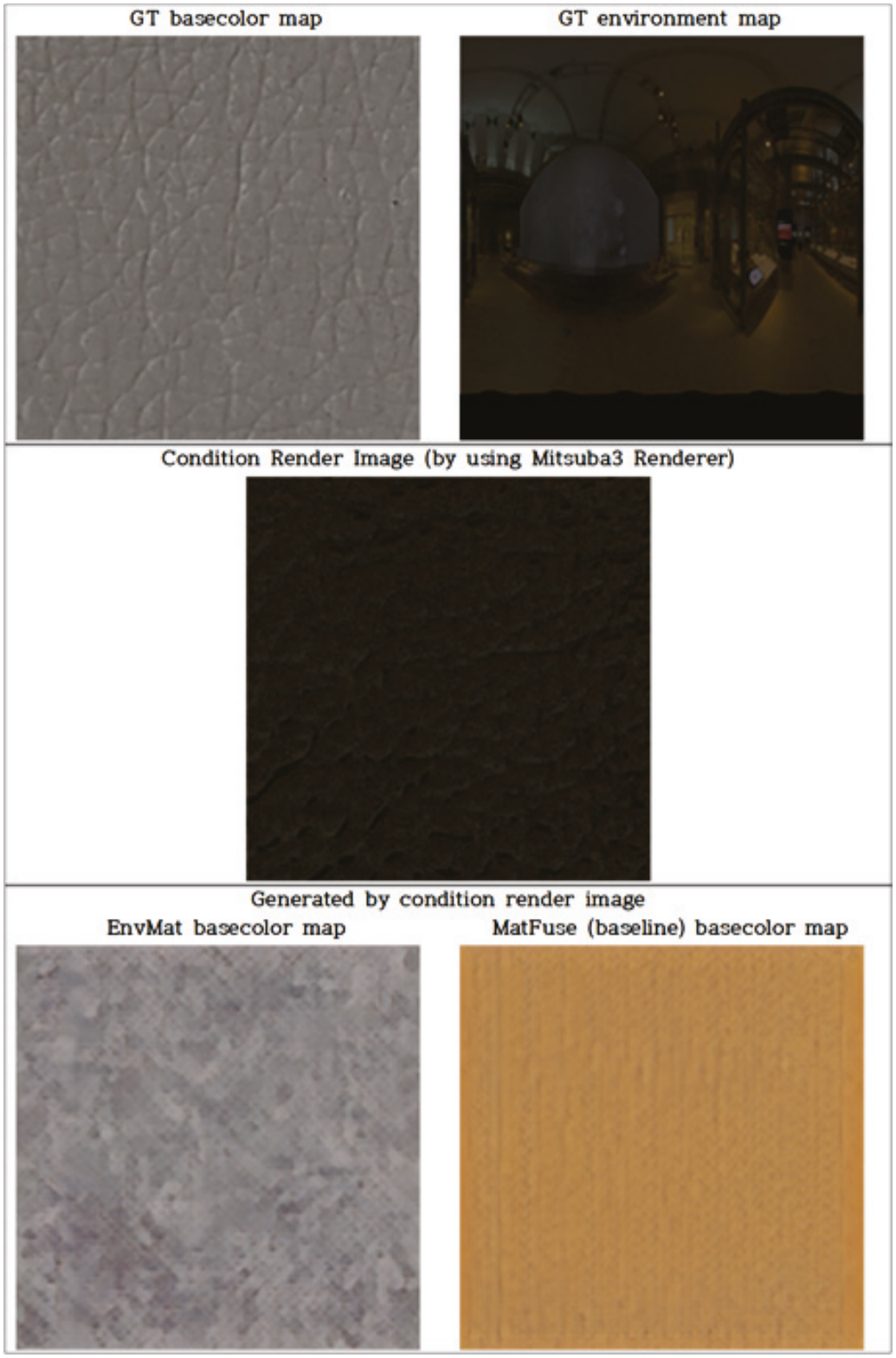


Figure A8. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

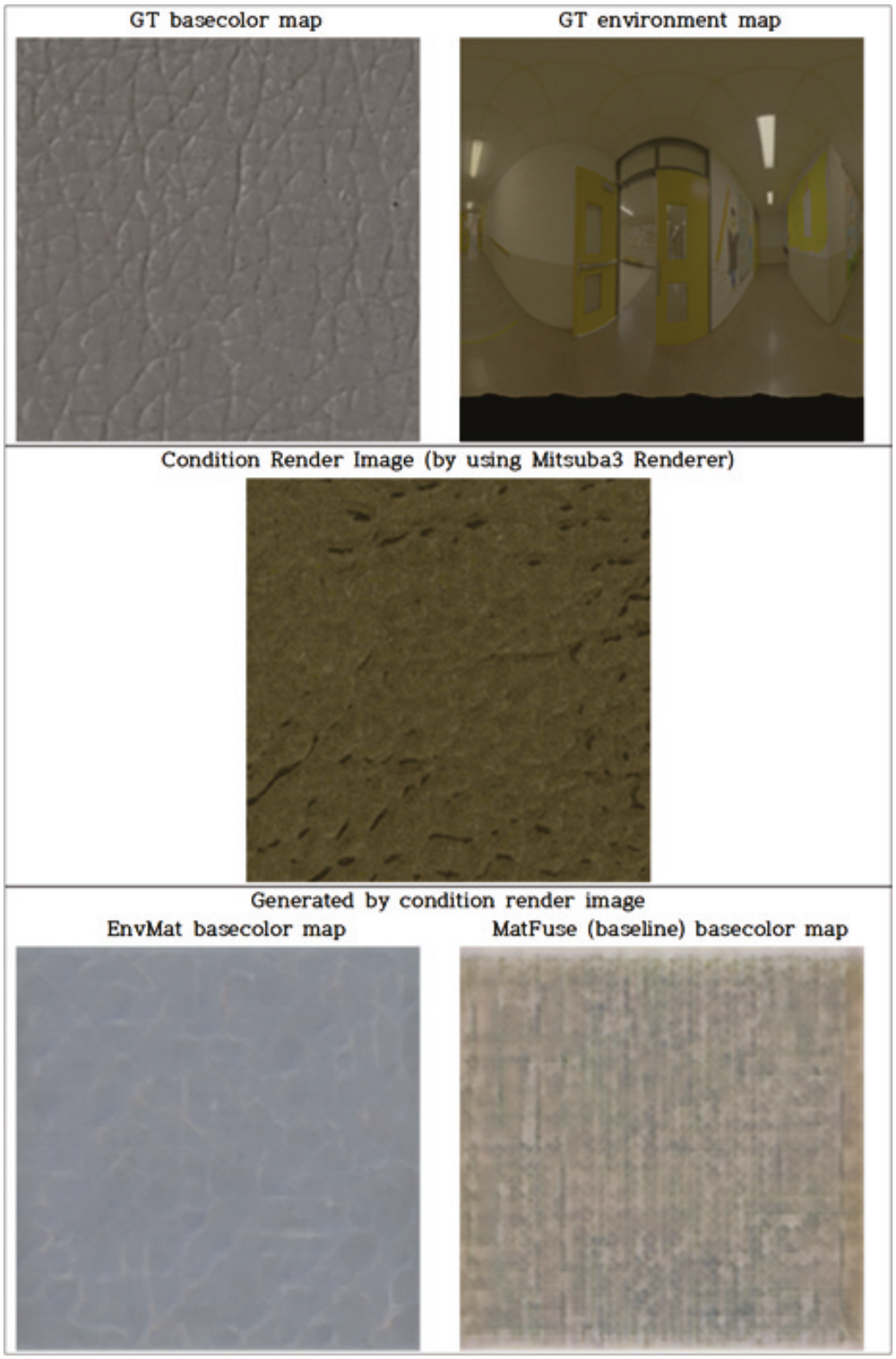


Figure A9. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

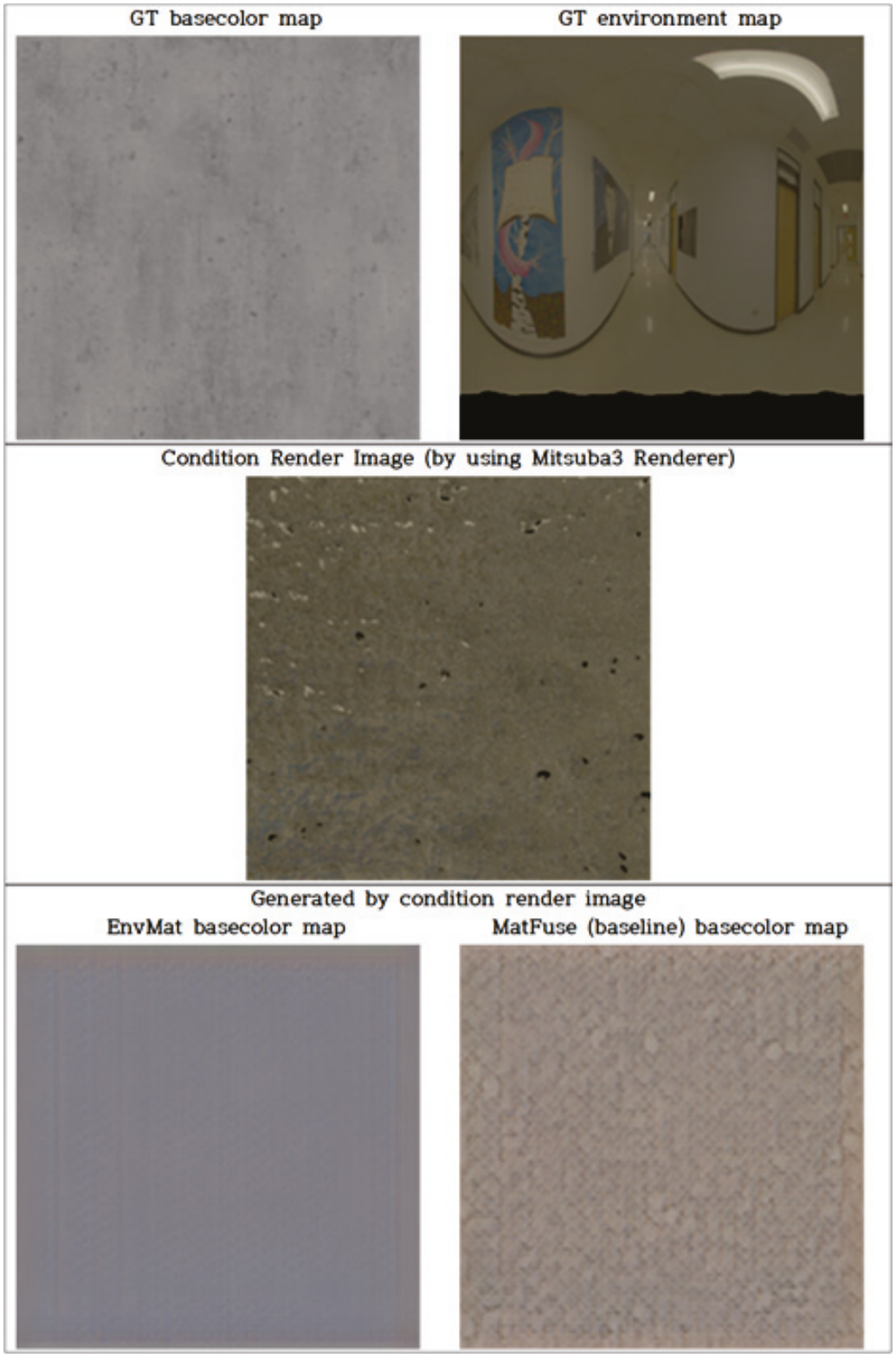


Figure A10. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

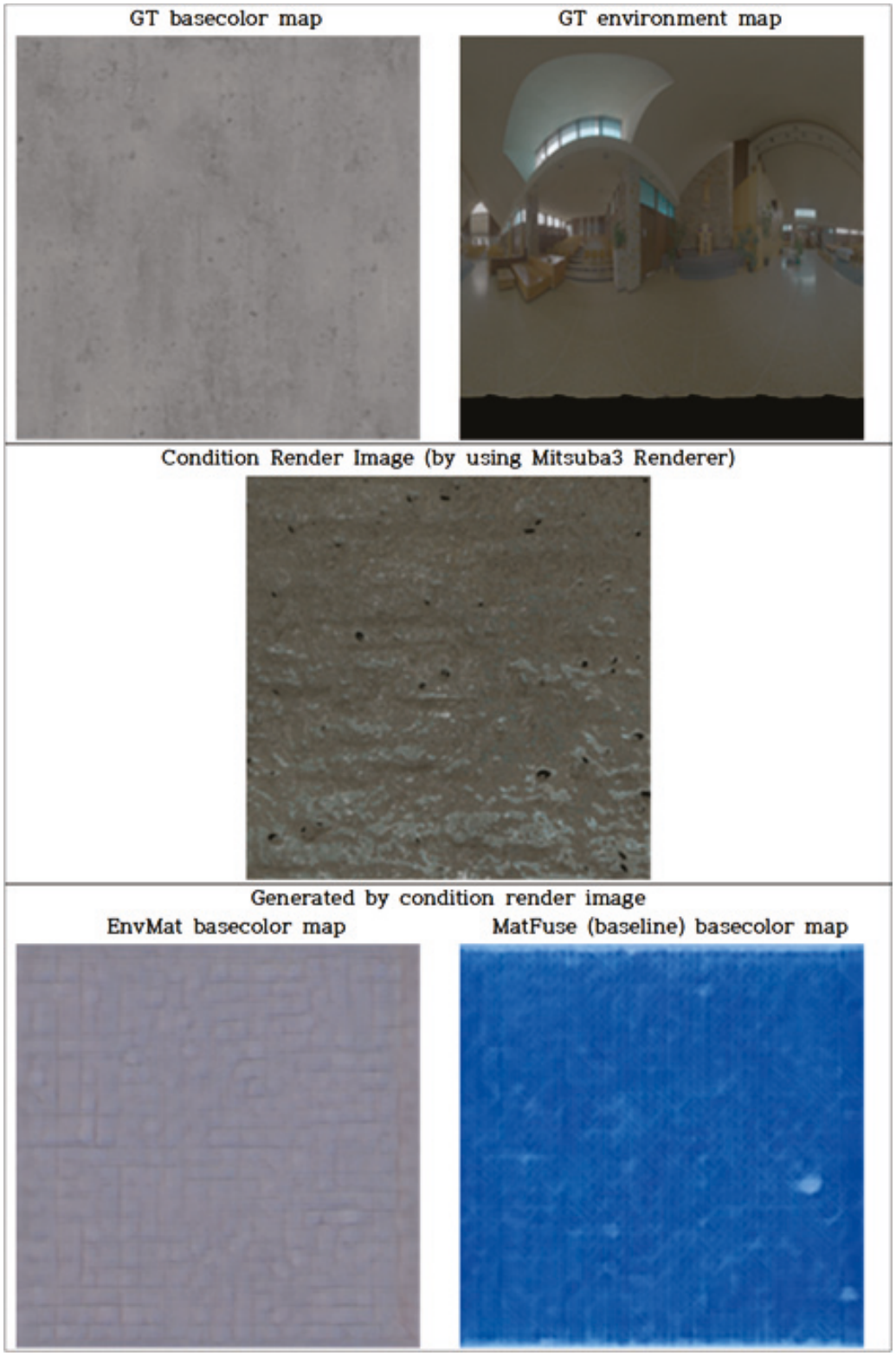


Figure A11. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

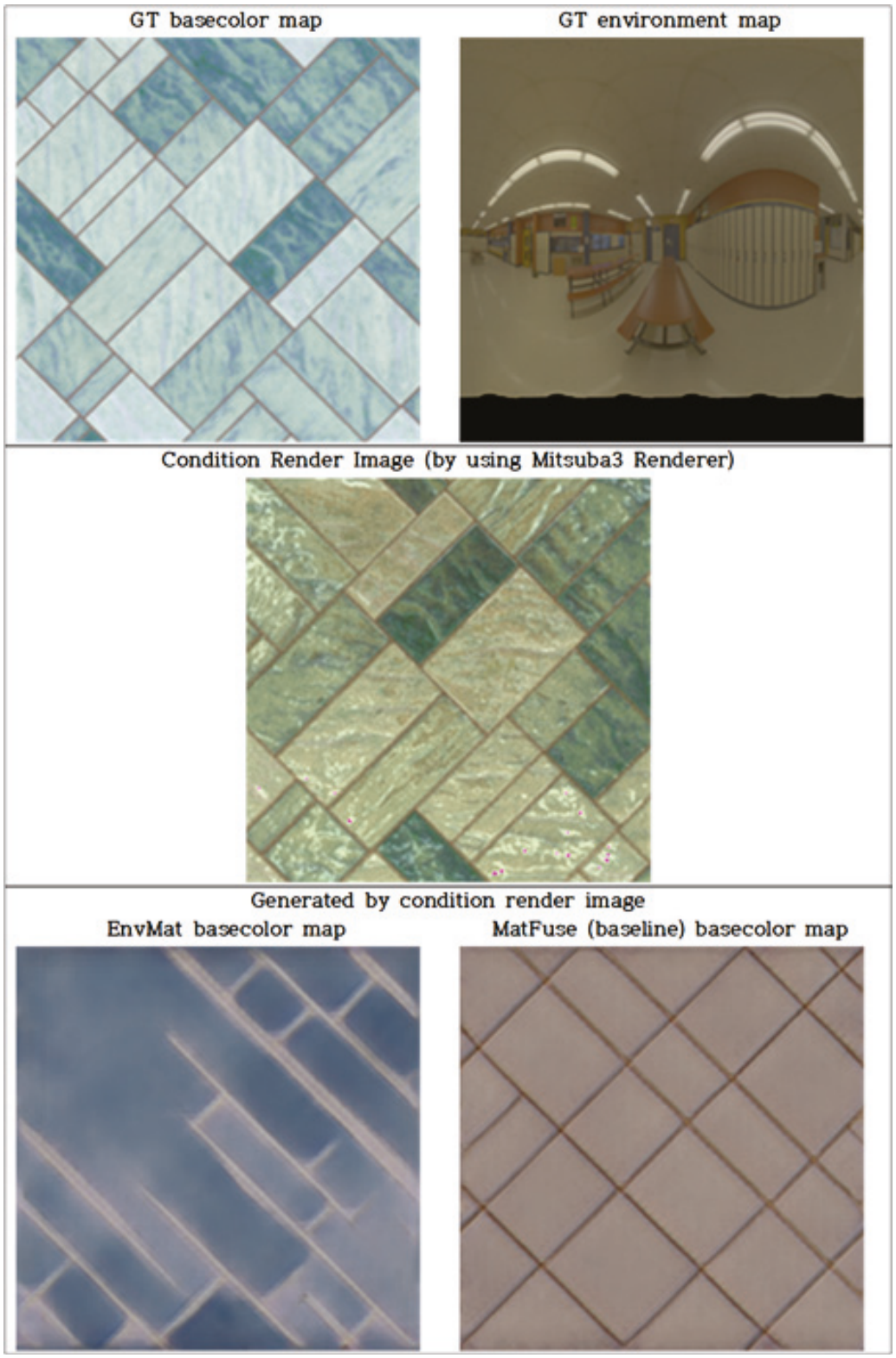


Figure A12. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

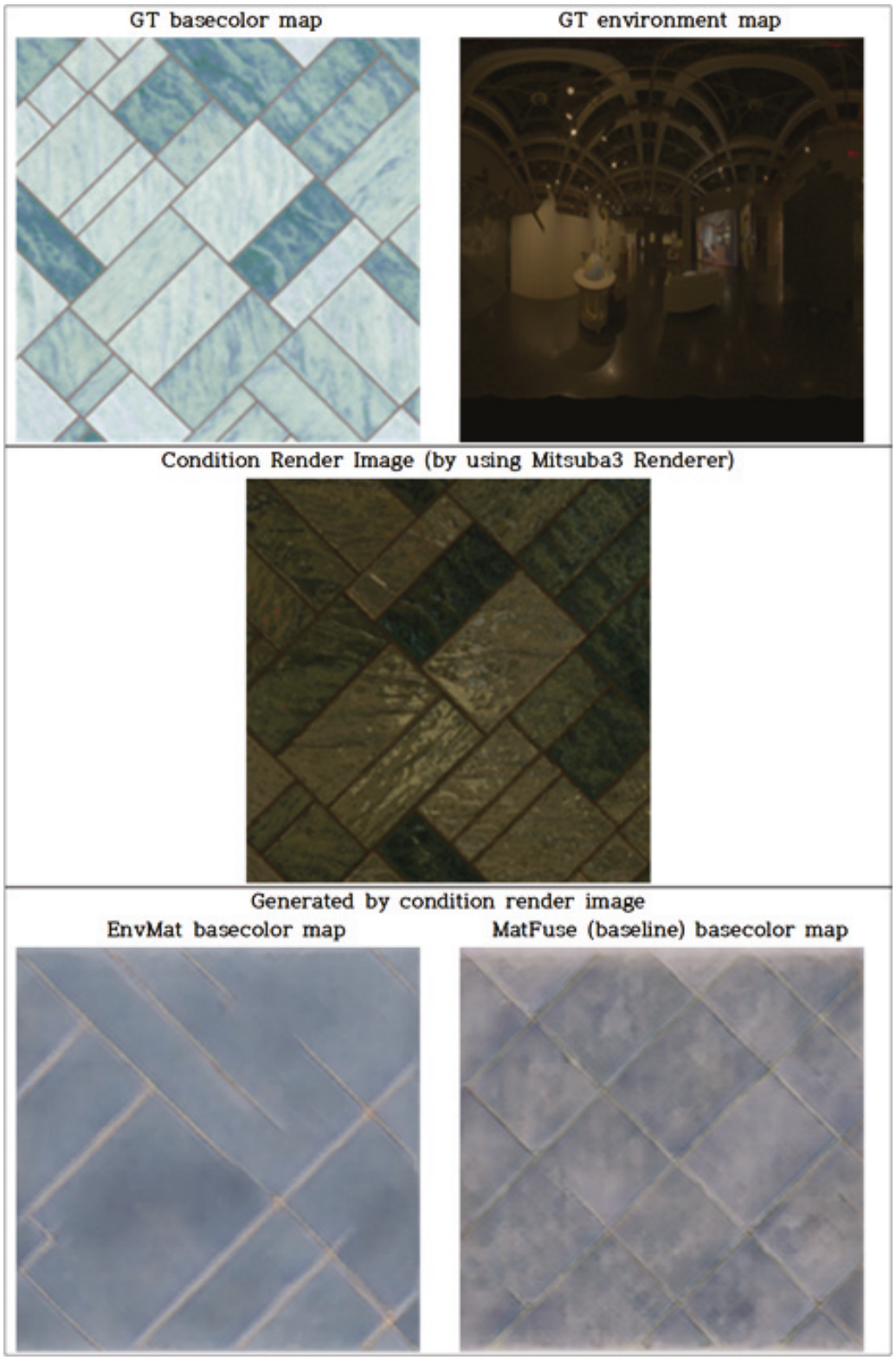


Figure A13. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

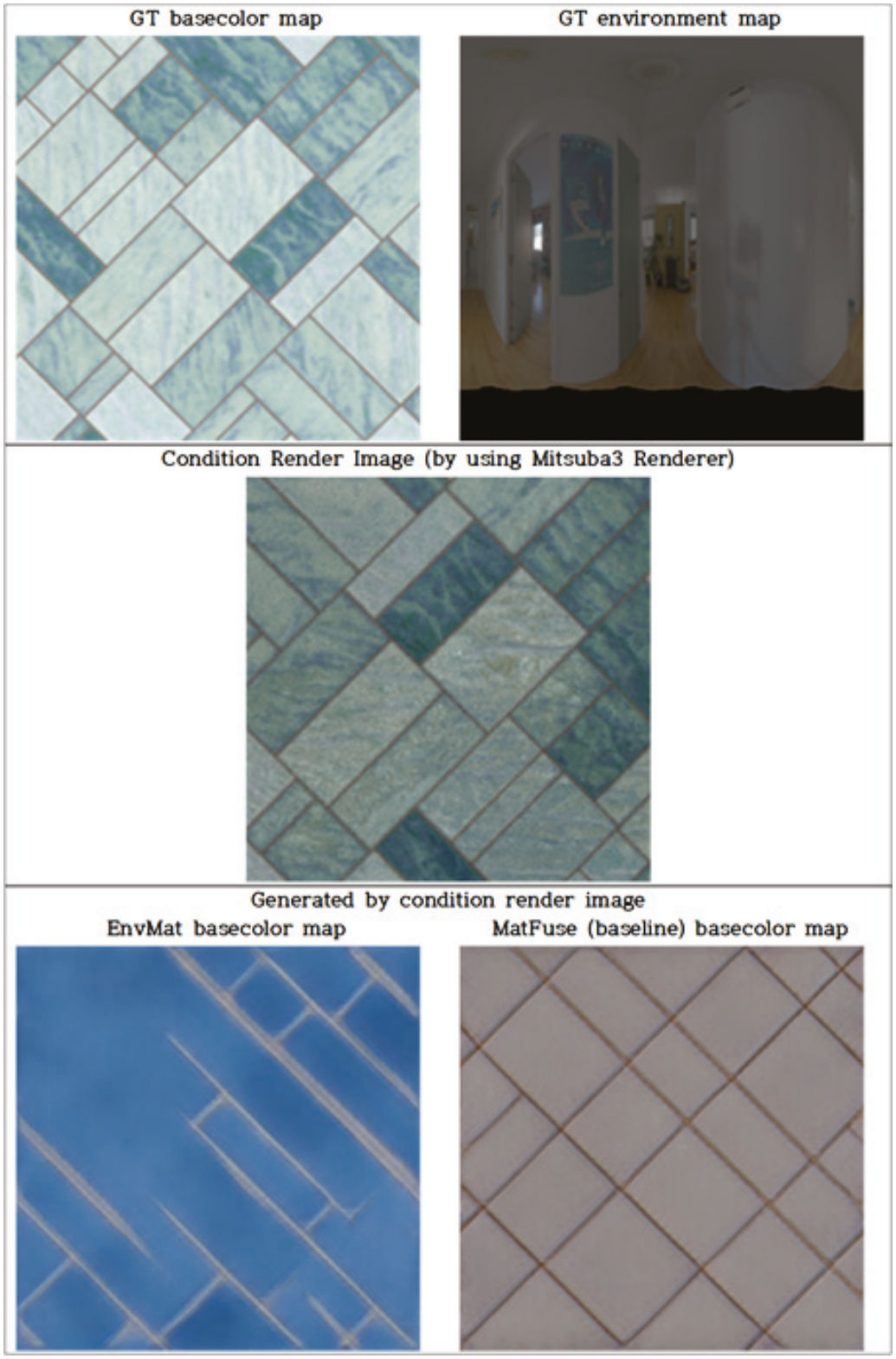


Figure A14. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

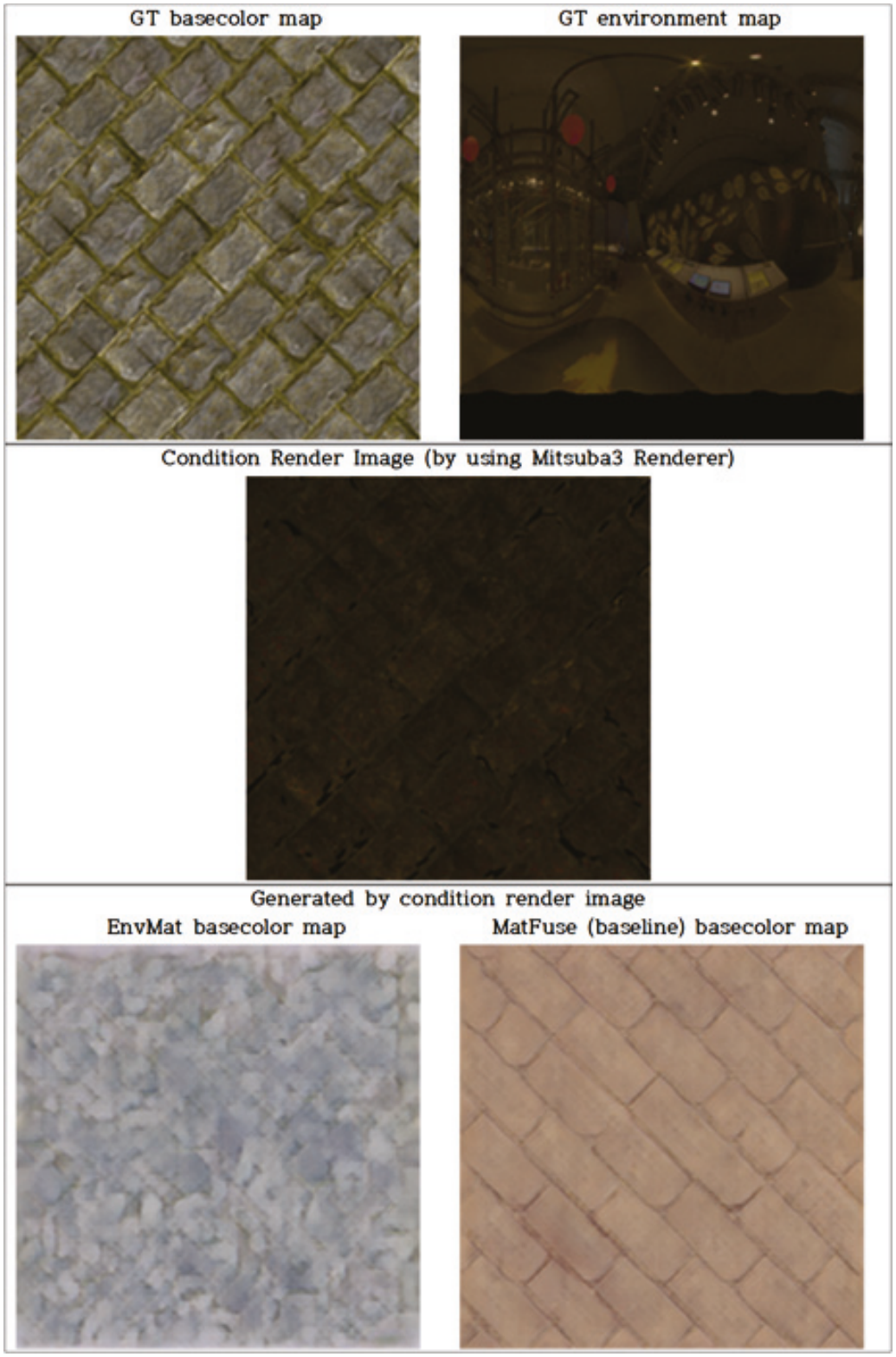


Figure A15. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

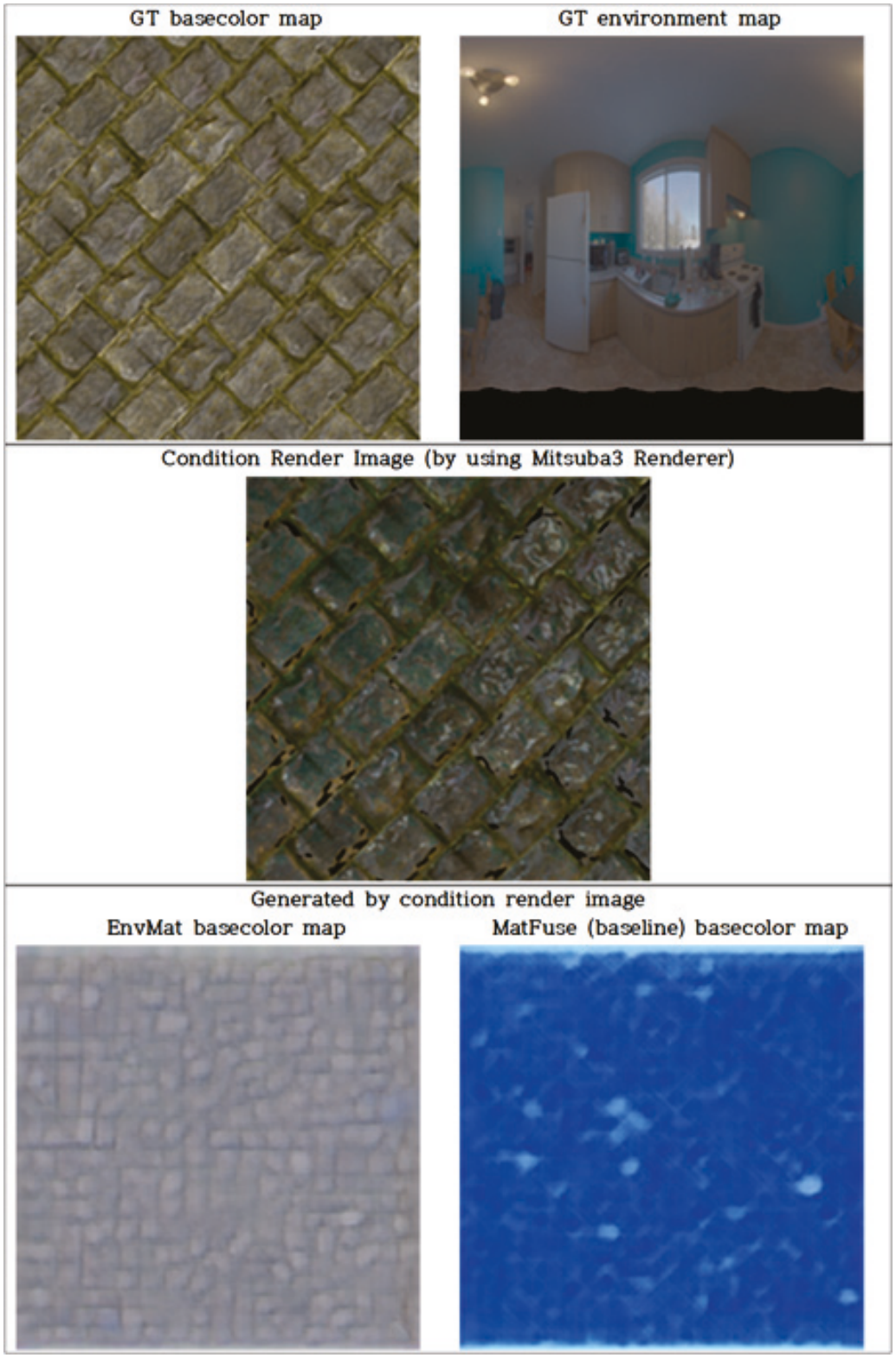


Figure A16. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

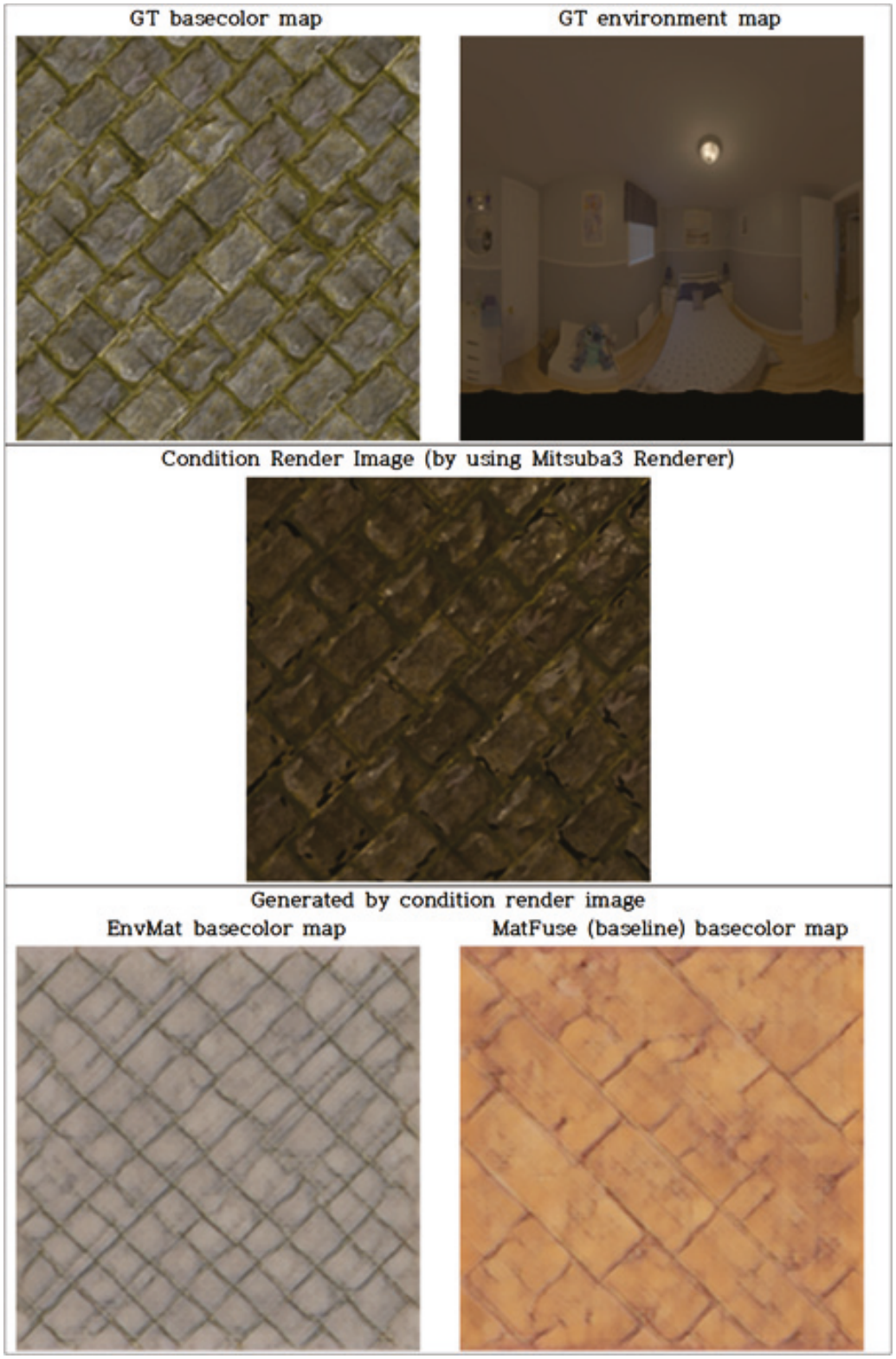


Figure A17. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

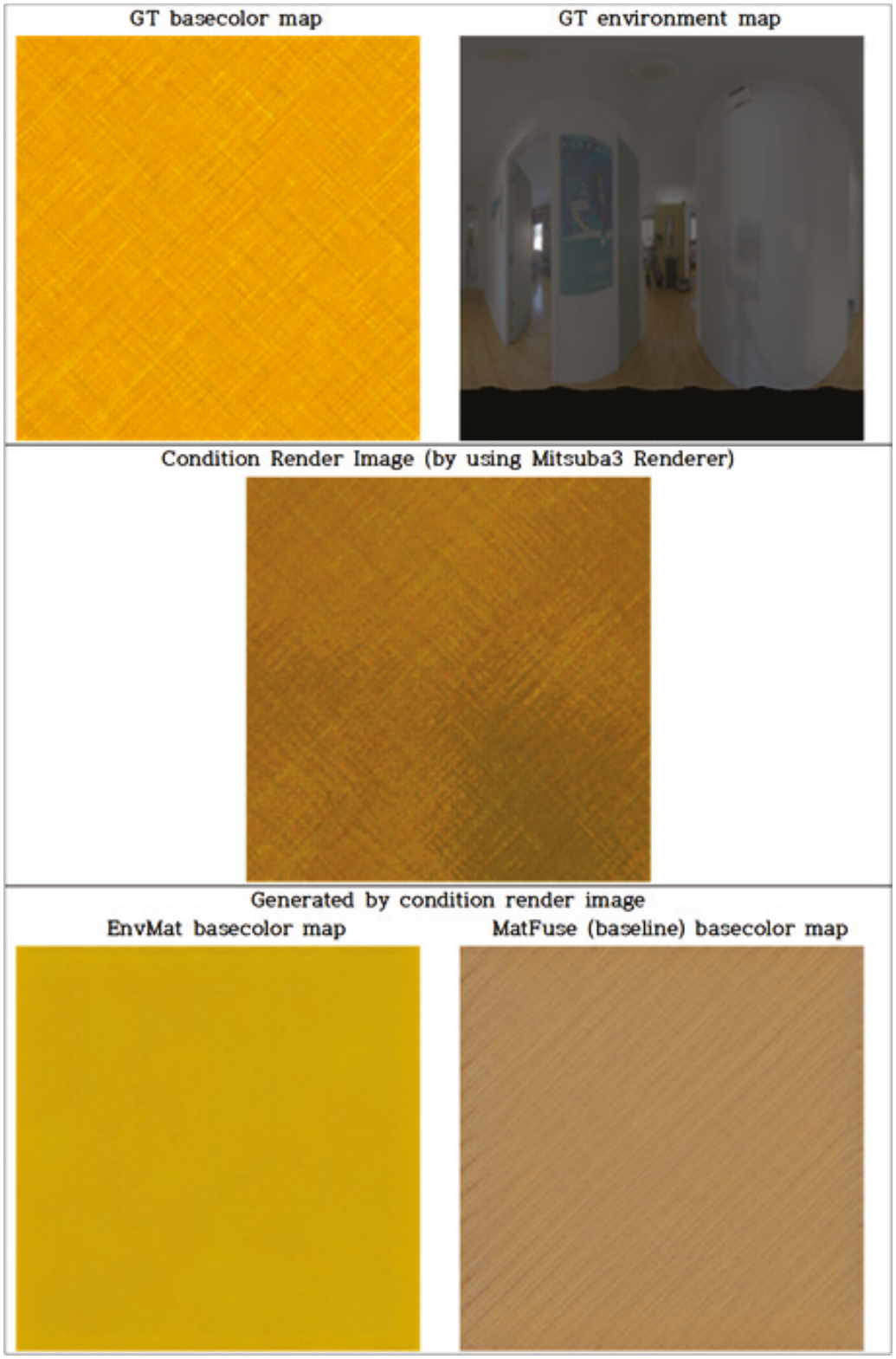


Figure A18. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

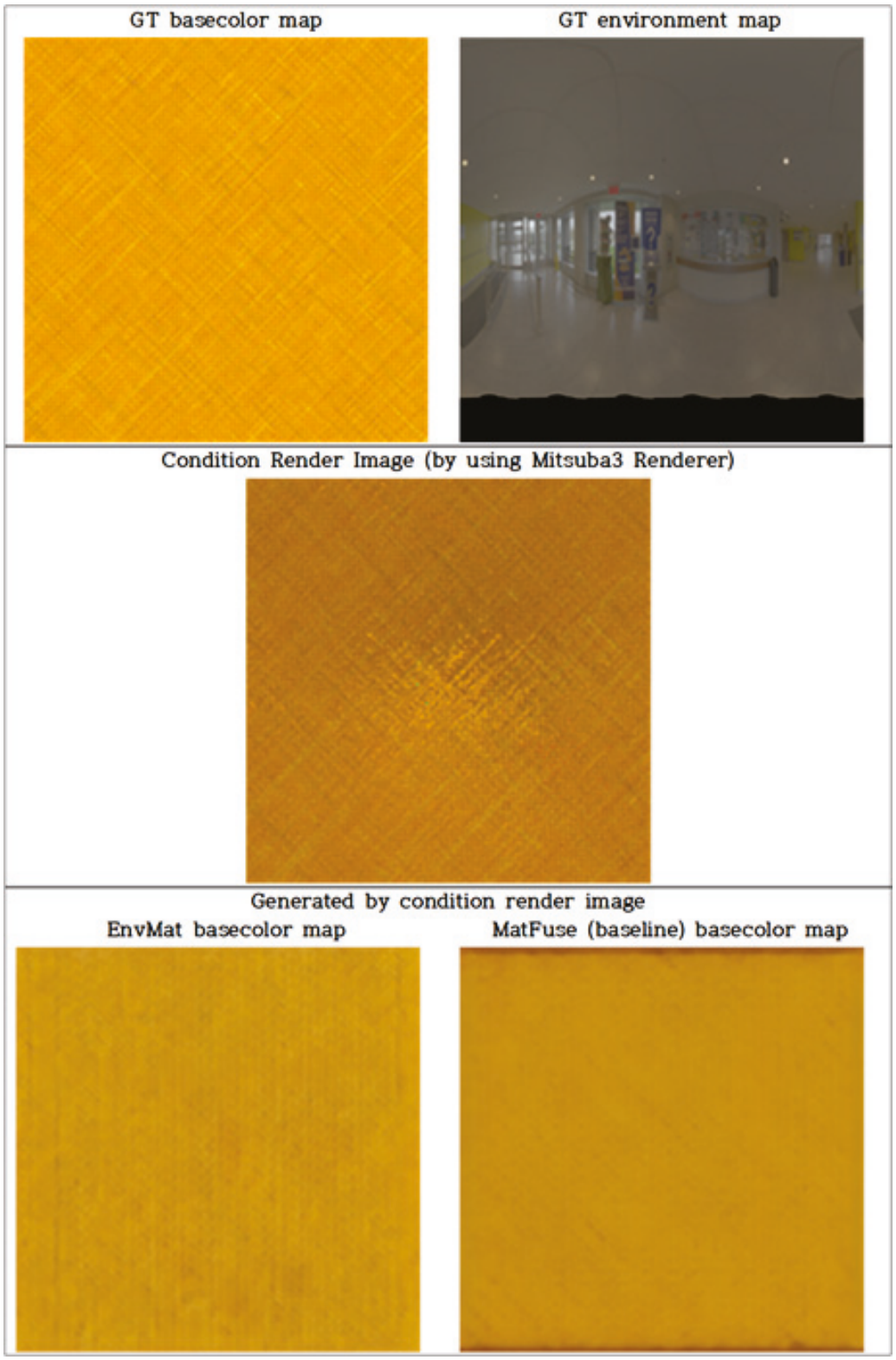


Figure A19. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

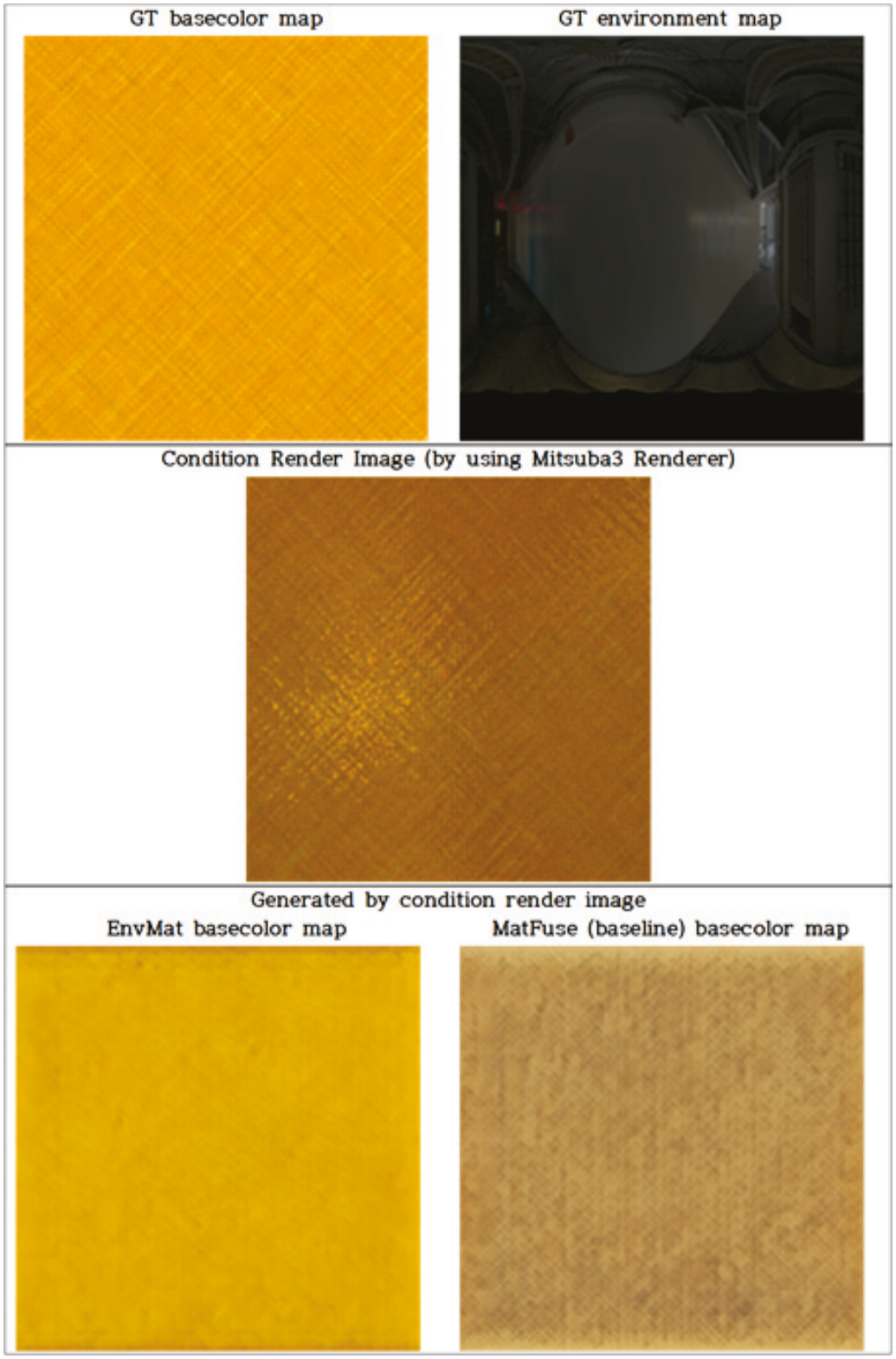


Figure A20. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

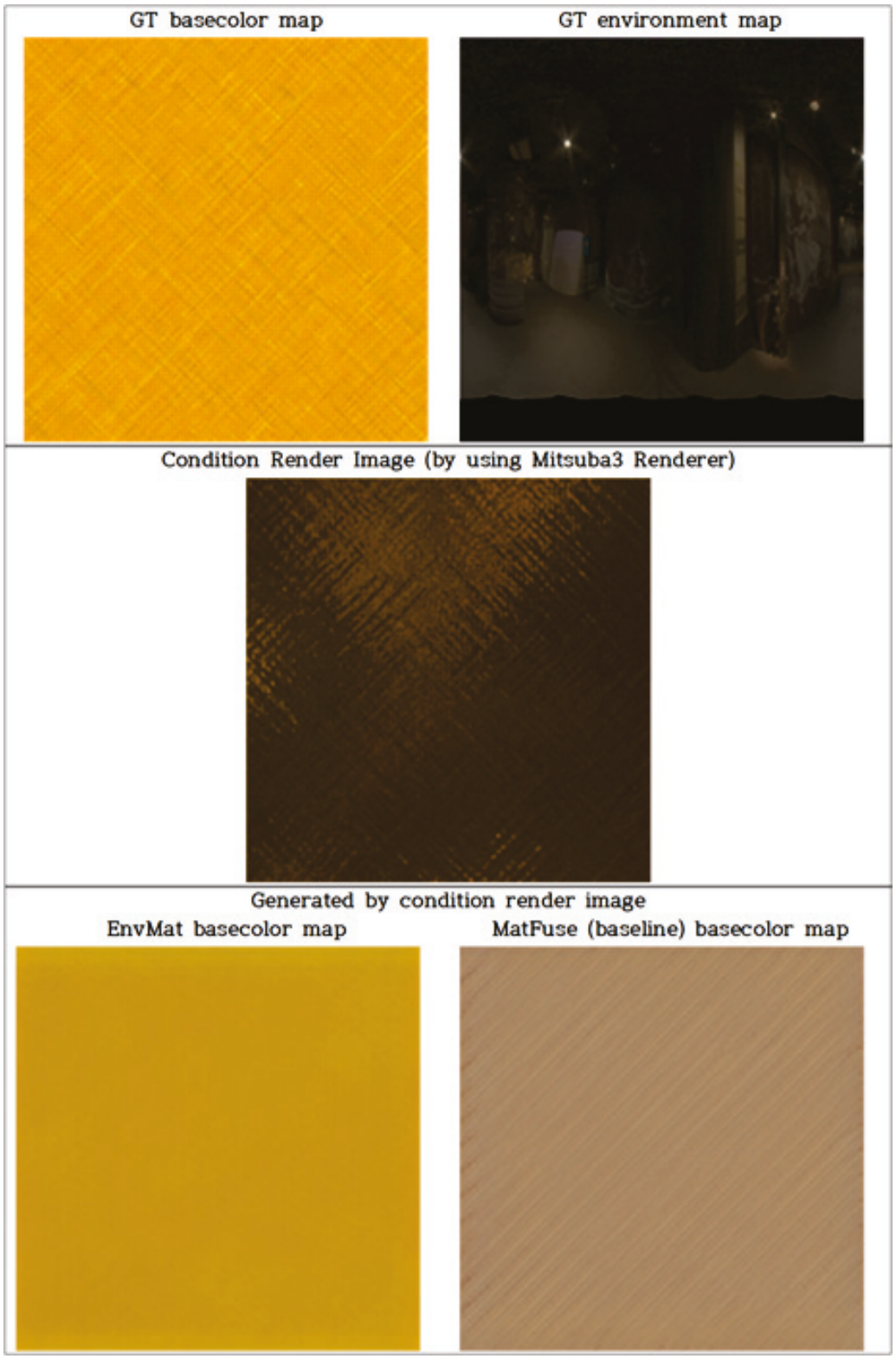


Figure A21. Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

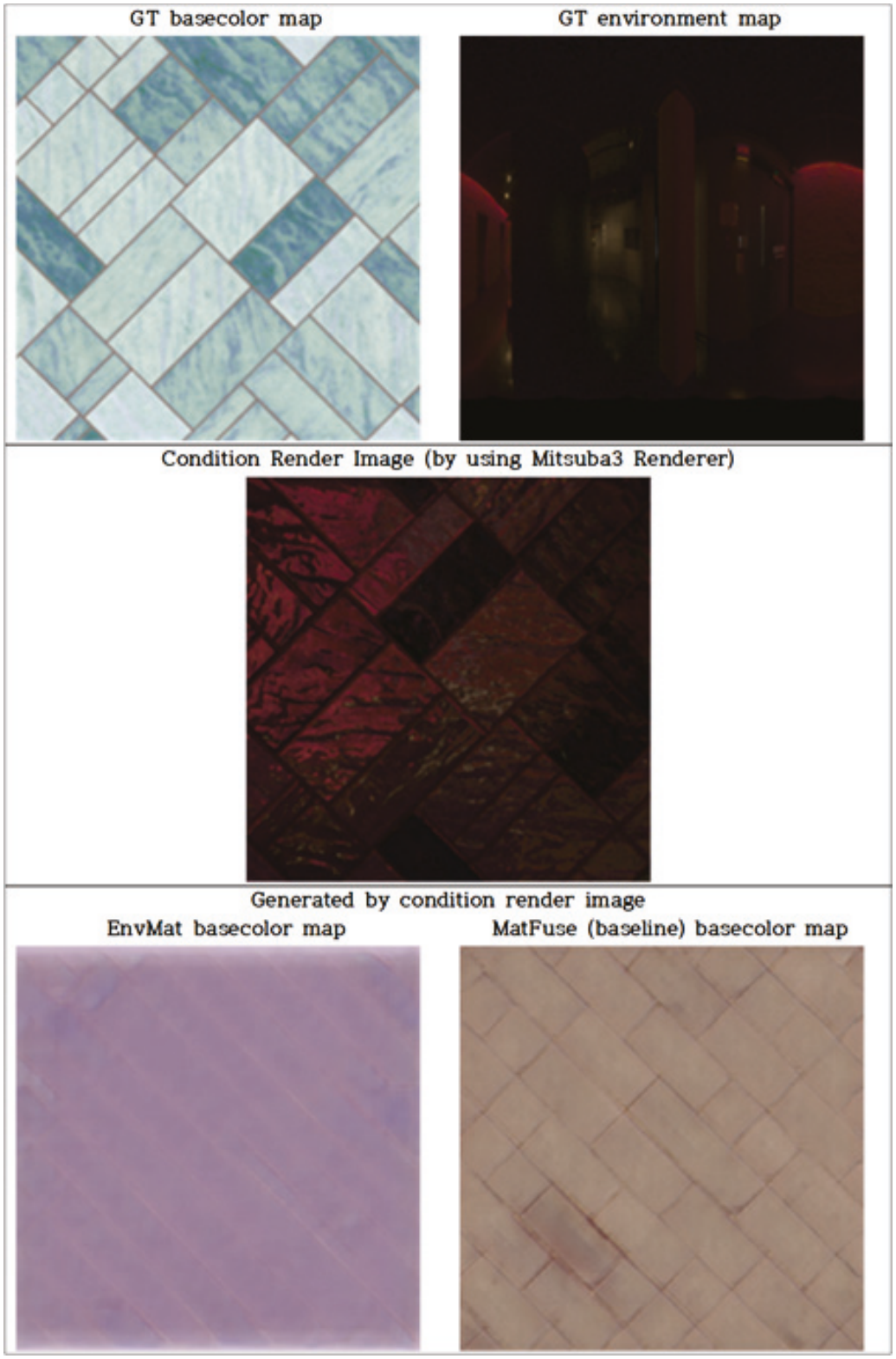


Figure A22. Limitations: Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

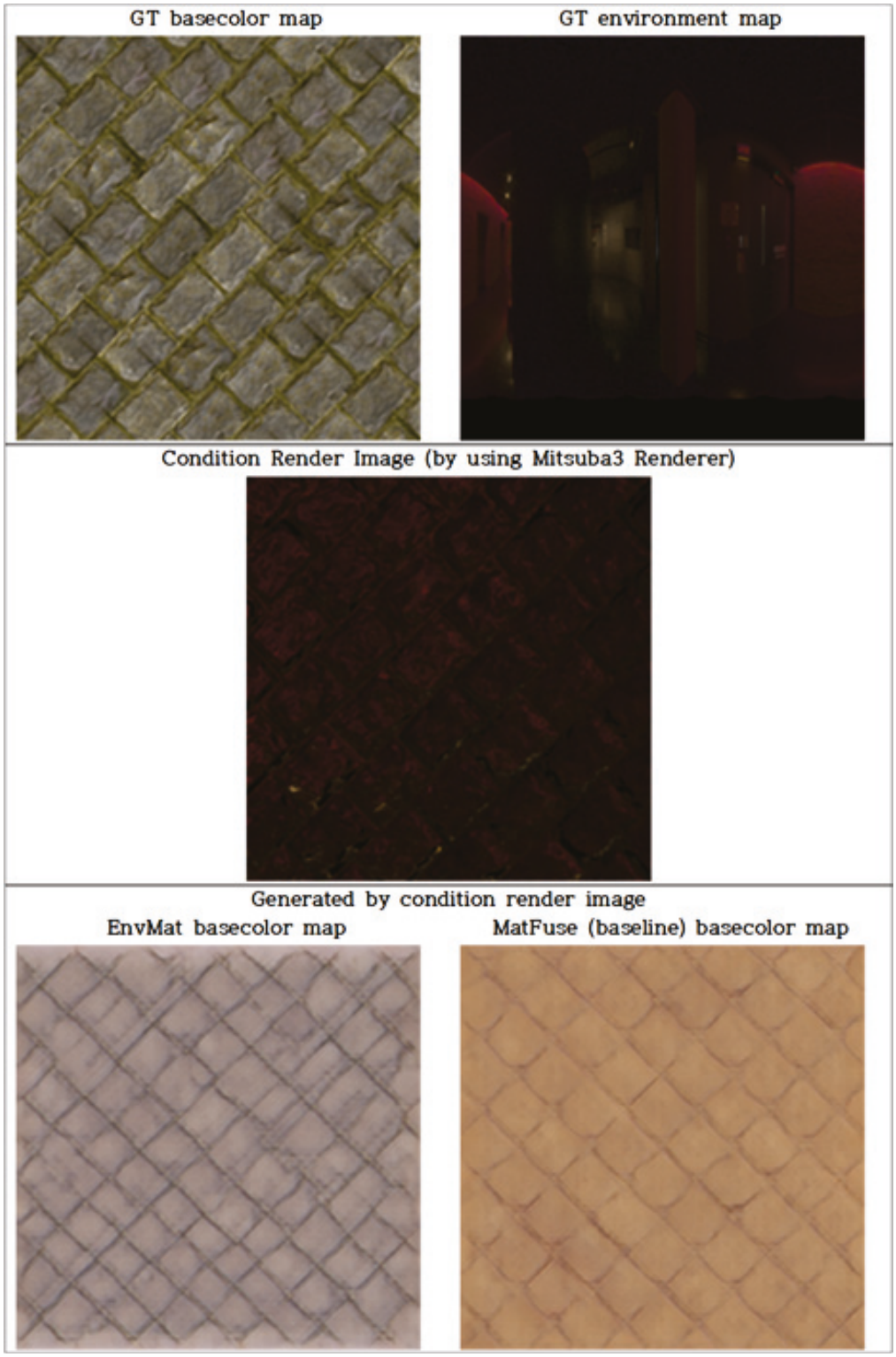


Figure A23. Limitations: Comparison of inference results of EnvMat (3 rows, 1 column) and MatFuse (3 rows, 2 columns). Inference results using the Condition render image (2 rows) generated from the same PBR map (1 row, 1 column) and env map (1 row, 2 columns) as input.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
2. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 8110–8119.

3. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
4. Vecchio, G.; Sortino, R.; Palazzo, S.; Spampinato, C. MatFuse: Controllable Material Generation with Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 4429–4438.
5. Deschaintre, V.; Aittala, M.; Durand, F.; Drettakis, G.; Bousseau, A. Single-Image SVBRDF Capture with a Rendering-Aware Deep Network. *ACM Trans. Graph.* **2018**, *37*, 128:1–128:15. [CrossRef]
6. Vecchio, G.; Palazzo, S.; Spampinato, C. SurfaceNet: Adversarial SVBRDF Estimation from a Single Image. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 Octpber 2021; pp. 12840–12848.
7. Guo, Y.; Smith, C.; Hašan, M.; Sunkavalli, K.; Zhao, S. MaterialGAN: Reflectance Capture Using a Generative SVBRDF Model. *ACM Trans. Graph.* **2020**, *39*, 254:1–254:13. [CrossRef]
8. Dhariwal, P.; Nichol, A. Diffusion Models Beat GANs on Image Synthesis. In Proceedings of the Advances in Neural Information Processing Systems 34, Virtual, 6–14 December 2021; pp. 8780–8794.
9. Esser, P.; Rombach, R.; Ommer, B. Taming Transformers for High-Resolution Image Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 12873–12883.
10. He, Z.; Guo, J.; Zhang, Y.; Tu, Q.; Chen, M.; Guo, Y.; Wang, P.; Dai, W. Text2Mat: Generating Materials from Text. In Proceedings of the Pacific Graphics 2023—Short Papers and Posters, Daejeon, Republic of Korea, 10–13 October 2023; pp. 89–97.
11. Martin, R.; Roullier, A.; Rouffet, R.; Kaiser, A.; Boubekeur, T. MaterIA: Single Image High-Resolution Material Capture in the Wild. *Comput. Graph. Forum* **2022**, *41*, 163–177. [CrossRef]
12. Vecchio, G.; Martin, R.; Roullier, A.; Kaiser, A.; Rouffet, R.; Deschaintre, V.; Boubekeur, T. ControlMat: A Controlled Generative Approach to Material Capture. *ACM Trans. Graph.* **2024**, *43*, 164:1–164:17. [CrossRef]
13. Vecchio, G.; Deschaintre, V. MatSynth: A Modern PBR Materials Dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 22109–22118.
14. Laval Indoor HDR Environment Map Dataset. 2014. Available online: http://hdrdb.com/indoor/ (accessed on 12 May 2025).
15. Chung, H.; Kim, J.; McCann, M.T.; Klasky, M.L.; Ye, J.C. Diffusion Posterior Sampling for General Noisy Inverse Problems. *arXiv* **2022**, arXiv:2209.14687.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Delving into YOLO Object Detection Models: Insights into Adversarial Robustness

Kyriakos D. Apostolidis and George A. Papakostas *

MLV Research Group, Department of Informatics, Democritus University of Thrace, 65404 Kavala, Greece; kyapostol@cs.duth.gr
* Correspondence: gpapak@cs.duth.gr

Abstract: This paper provides a comprehensive study of the security of YOLO (You Only Look Once) model series for object detection, emphasizing their evolution, technical innovations, and performance across the COCO dataset. The robustness of YOLO models under adversarial attacks and image corruption, offering insights into their resilience and adaptability, is analyzed in depth. As real-time object detection plays an increasingly vital role in applications such as autonomous driving, security, and surveillance, this review aims to clarify the strengths and limitations of each YOLO iteration, serving as a valuable resource for researchers and practitioners aiming to optimize model selection and deployment in dynamic, real-world environments. The results reveal that YOLOX models, particularly their large variants, exhibit superior robustness compared to other YOLO versions, maintaining higher accuracy under challenging conditions. Our findings serve as a valuable resource for researchers and practitioners aiming to optimize YOLO models for dynamic and adversarial real-world environments while guiding future research toward developing more resilient object detection systems.

Keywords: computer vision; YOLO; deep learning; adversarial robustness; object detection

1. Introduction

Object detection is a pivotal task in computer vision, integral to a wide range of applications including autonomous vehicles [1], robotics [2], surveillance [3], and various fields like medicine [4] and agriculture [5]. Specifically, YOLO models have been successfully applied in medicine for tasks such as the localization and segmentation of lumbar spine vertebrae deformities [6] and detection of breast lesions [7]. In agriculture, YOLO has been employed for roadside monitoring and the detection of invasive alien plant species [8], as well as the real-time detection of invasive weed seedlings like Solanum rostratum Dunal [9]. Among the numerous algorithms developed for real-time object detection, YOLO (You Only Look Once) stands out due to its exceptional balance between accuracy and speed. Introduced by Redmon et al. [10] in 2015, YOLOv1 marked a significant advancement in the field. Since then, the YOLO algorithm has undergone continuous evolution, with new versions being released almost annually, culminating in the latest iteration, YOLOv11 (Figure 1).

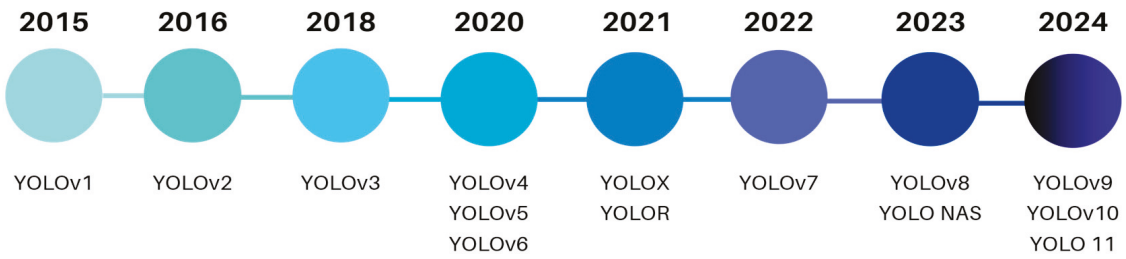


Figure 1. Evolution of YOLO models.

Apart from YOLO, various object detectors have been proposed in recent years. R-CNN (region-based convolutional neural network) [11] is a pioneering model that uses selective search to propose regions and then classifies each region using a convolutional neural network. Fast R-CNN [12] is an improved version of R-CNN that integrates region proposal and classification into a single network, significantly speeding up the detection process. Furthermore, Faster R-CNN [13] advances Fast R-CNN by introducing a region proposal network (RPN) that shares features with the detection network, enabling end-to-end training and real-time performance. Another significant object detector is SSD (Single Shot MultiBox Detector) [14], which predicts bounding boxes and class scores for multiple objects in a single forward pass, facilitating real-time detection. Additionally, EfficientDet [15] is a family of object detection models that use EfficientNet [16] as the backbone and employ a compound scaling method to achieve a balance between accuracy and efficiency. Lastly, DETR (Detection Transformer) [17] is a novel model that leverages transformers for end-to-end object detection, eliminating the need for traditional components like region proposal networks and non-maximum suppression, and achieving competitive accuracy with a simpler architecture.

Modern deep neural networks (DNNs) are integral to the success of the aforementioned algorithms. However, numerous studies have shown that DNNs are susceptible to adversarial attacks involving imperceptible perturbations and noisy data [18–20]. This vulnerability poses a significant challenge to the deployment of object detectors in real-world conditions, potentially compromising their reliability and effectiveness. Adversarial attacks, which involve adding subtle perturbations to input images, can significantly degrade the performance of deep learning models, leading to incorrect predictions. These attacks exploit the vulnerabilities inherent in deep networks, raising critical concerns about the security of YOLO models in safety-sensitive applications. Alongside adversarial attacks, real-world image corruptions—such as noise, blur, and weather-induced distortions—further impact model performance. As such, robustness to these disruptions is crucial for deploying YOLO models in environments where pristine images are not guaranteed.

Several reviews on YOLO models exist, including [21–23], yet none have thoroughly examined their effectiveness under noisy data and adversarial examples. This work aims to address this gap by evaluating the robustness of YOLO model series under Fast Gradient Sign Method (FGSM) [24] and Projected Gradient Descent (PGD) [25] attacks, along with other image corruption techniques, to test their resilience. By analyzing the behavior of these models across different scenarios, we seek to identify the strengths and weaknesses of each YOLO variant. Our findings provide insights into the vulnerabilities of YOLO models, guiding future research towards more resilient object detection systems and informing practitioners on model suitability under challenging real-world conditions.

The remainder of this paper is organized as follows: Section 2 presents an overview of existing works that address robustness in YOLO algorithms, highlighting key approaches and gaps in the current literature. Section 3 provides a detailed overview of the dataset, experimental setup, and methodologies applied to evaluate the robustness of YOLO models

against adversarial attacks and image corruptions. In Section 4, we discuss the evolution of the YOLO architectures, highlighting key advancements and innovations across each version. Section 5 presents our experimental results, analyzing the robustness of YOLO models under adversarial perturbations and image corruptions. In Section 6, we provide a comparative discussion of the findings and insights into the models' resilience. Finally, Section 7 concludes the paper by summarizing our contributions and suggesting future research directions.

2. Robustness in YOLO Algorithms

Recent research has increasingly focused on assessing and improving the robustness of YOLO-based object detectors, particularly under adversarial attacks and visual corruptions. Several studies have revealed the susceptibility of YOLO models to a wide range of digital adversarial attacks, such as FGSM, PGD, and C&W, from Jain [26], demonstrating that YOLOv5 can be easily misled in traffic sign detection. To better exploit YOLO's structure, Dai et al. [27] proposed an Adaptive Deformation Method (ADM) targeting bounding box predictions, achieving higher attack success rates on YOLOv4 and YOLOv5. In parallel, Bayer et al. [28] introduced "eigenpatches"—PCA-derived adversarial patches—that efficiently fool YOLOv7 with reduced computational cost. Addressing the physical domain, Schack et al. [29] and Lian et al. [30] emphasized the real-world limitations of patch attacks and provided comprehensive benchmarks (e.g., PADetBench) to evaluate robustness under realistic environmental transformations. Beyond adversarial attacks, several works evaluated YOLO under image corruptions. Liu et al. [31] benchmarked detectors on COCO-C and BDD100K-C, showing that YOLO models generally handle common corruptions better than two-stage counterparts. In aerial vision, Arsenos et al. [32] found YOLOv5 and YOLOX to be particularly robust under simulated fog, rain, and blur, while Ding and Luo [33] introduced SDNIA-YOLO, enhancing detection under extreme weather using image-adaptive enhancements and stylized data augmentation. Collectively, these studies highlight both the vulnerabilities and promising resilience of YOLO models in adverse visual conditions, motivating further research into robustness-aware training and evaluation.

3. Materials and Methods

Ensuring the robustness of object detection models is crucial for real-world applications, particularly in safety-critical domains such as autonomous driving and surveillance. This study evaluates the resilience of various YOLO models against adversarial attacks and image corruptions, aiming to identify the most robust architectures. To achieve this, we employed a standardized experimental setup, using the COCO dataset and a diverse selection of YOLO versions ranging from YOLOv3 to YOLOv11, including various model sizes (n, s, m, l, x). These models were selected to cover the full spectrum of advancements in the YOLO framework, from early iterations to the most recent state-of-the-art versions. By leveraging pre-trained weights from official repositories, we ensured that all models were tested in their standard configurations, allowing for a fair comparison. Testing multiple model sizes also provided insight into how architectural complexity influences robustness. Figure 2 shows the pipeline we followed for the experiments and the robustness evaluation. The following sections describe the dataset, attack methodologies, and evaluation metrics used to systematically assess each model's resilience under different perturbation scenarios.

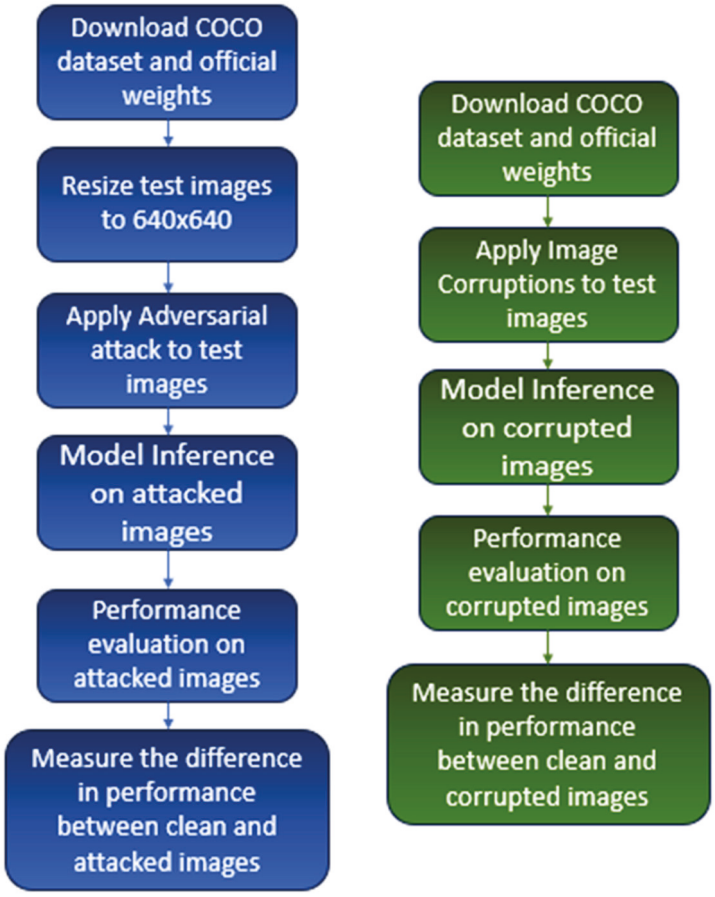


Figure 2. Pipeline for robustness evaluation. (**a**) Adversarial attacks, (**b**) image corruption.

3.1. Evaluation Metrics and Dataset

In object detection tasks, the accurate evaluation of a model's performance is critical for understanding its ability to identify and localize objects within an image. Two key metrics often employed for this purpose are Intersection over Union (IoU) and Average Precision (AP). IoU measures the overlap between the predicted and actual locations of objects, providing a quantitative assessment of localization accuracy. It serves as a fundamental criterion for determining how well a detection model aligns its predictions with the ground truth. On the other hand, AP offers a comprehensive measure of a model's detection capabilities by considering the trade-off between precision and recall across varying confidence thresholds. Together, IoU and AP help in evaluating both the accuracy of object localization and the quality of object recognition, making them integral to benchmarking object detection models.

In YOLOv1 and YOLOv2, the PASCAL VOC 2007 [34] and VOC 2012 datasets [35] were employed for training and benchmarking purposes. Starting with YOLOv3, however, the Microsoft COCO (Common Objects in Context) [36] dataset was utilized.

To ensure a fair evaluation of the models, it is essential to assess them using the same dataset. Given that YOLOv1 and YOLOv2 were trained and evaluated on different datasets, they are excluded from this evaluation. We conducted the attacks on the COCO dataset, which is utilized by YOLOv3 and the subsequent versions.

AP and IoU Explanation

Average Precision (AP) is a widely used metric for evaluating the performance of classification models, particularly in object detection and information retrieval tasks. It is

calculated as the area under the Precision–Recall (PR) curve, which represents the trade-off between precision and recall across different classification thresholds.

Precision is defined as the ratio of true positive predictions to the total number of positive predictions made by the model, capturing the accuracy of positive predictions. It is given by the following:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} \tag{1}$$

Recall measures the proportion of actual positive instances that are correctly identified by the model. It is calculated as follows:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \tag{2}$$

The Precision–Recall curve is generated by plotting precision against recall while varying the decision threshold of the model. As the threshold changes, the model's predictions become more or less conservative, resulting in different precision and recall values. AP is computed as the integral of the Precision–Recall curve, which effectively measures the area under this curve (AUC). This can be expressed as follows:

$$AP = \sum_{n} (R_n - R_{n-1}) * P_n \tag{3}$$

where *Pn* is the precision at the *n*-th threshold, *Rn* is the recall at the *n*-th threshold. Mean **Average Precision (mAP)** extends this concept by averaging the AP values across multiple classes. It is particularly useful when evaluating multi-class classification tasks, where the performance of each class is important. Given *C* classes, mAP is calculated as follows:

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \tag{4}$$

where AP*i* represents the Average Precision for the *i*-th class.

This metric provides a balanced assessment of the model's performance across all classes by considering both precision and recall. It is especially helpful in applications where false positives (which affect precision) and false negatives (which affect recall) have different impacts on the overall system. By summarizing the area under the Precision–Recall curve, AP and mAP capture the model's ability to maintain high precision while achieving high recall.

Intersection over Union (IoU) is a widely used metric for evaluating the localization accuracy of object detection models. It is especially effective for assessing how closely a predicted bounding box aligns with the ground truth bounding box for a given object. IoU provides a quantitative measure of the overlap between two regions, offering insights into the localization errors in object detection.

To compute the IoU between a predicted bounding box *Bp* and a ground truth bounding box *Bg*, we follow these steps:

Intersection: The intersection area is defined as the region where the predicted bounding box overlaps with the ground truth bounding box. This area is computed by identifying the coordinates that form the overlapping region between *Bp* and *Bg*.

Union: The union area is the total area covered by both the predicted and ground truth bounding boxes, without double-counting the overlapping region. It represents the sum of the areas of *Bp* and *Bg* minus the intersection area.

IoU Calculation: IoU is calculated as the ratio of the intersection area to the union area.

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} = \frac{|B_p\ \cap B_g|}{|B_p\ \cup B_g|} \tag{5}$$

where $|Bp \cap Bg|$ denotes the area of the intersection between the predicted and ground truth bounding boxes, and $|Bp \cup Bg|$ represents the area of the union.

The resulting IoU score ranges from 0 to 1, where:

- An IoU of 1 indicates perfect alignment between the predicted and ground truth bounding boxes.
- An IoU of 0 indicates no overlap between the two bounding boxes.
- Higher IoU values indicate better localization accuracy, while lower values signify larger discrepancies between the predicted and actual bounding boxes.

IoU is commonly used as a threshold-based metric in object detection models. For instance, predictions are often considered accurate if they achieve an IoU above a specified threshold (e.g., 0.5). By providing a reliable measure of how closely a model's predictions match the actual object locations, IoU serves as a crucial criterion for model evaluation and comparison in tasks involving object detection and localization.

While these metrics provide a fundamental understanding of object detection performance, real-world conditions often introduce challenges such as adversarial perturbations and environmental corruption. In the following section, we examine the impact of these factors on YOLO models.

3.2. Image Attacks

3.2.1. Adversarial Attacks

In the realm of adversarial machine learning, the robustness of neural networks is frequently tested using adversarial attacks, which are carefully crafted perturbations added to the input data to deceive the model into making incorrect predictions. Among the various techniques developed to generate adversarial examples, FGSM and PGD are two of the most prominent and widely studied methods. These attacks not only reveal vulnerabilities in machine learning models but also help in understanding and improving model robustness. In the following sections, we delve into the mechanics of FGSM and PGD attacks, examining their methodologies and impact on model performance.

FGSM is a simple yet effective technique for generating adversarial examples, as introduced by Goodfellow et al. in 2014 [24]. The core idea behind FGSM is to perturb the input data in a direction that maximizes the loss of the model. This is achieved by adding a small, carefully calculated noise to the input data, which is computed based on the gradient of the loss with respect to the input. Given an input x, a true label y, a model θ, and a loss function $J(\theta,x,y)$, the adversarial example x' is generated as follows:

$$x' = x + \epsilon * sign(\nabla x J(\theta, x, y)) \tag{6}$$

Here, ϵ is a small scalar value representing the perturbation magnitude, and sign($\nabla xJ(\theta,x,y)$) is the sign of the gradient of the loss function with respect to the input. Figure 3 shows the impact of ϵ on images.

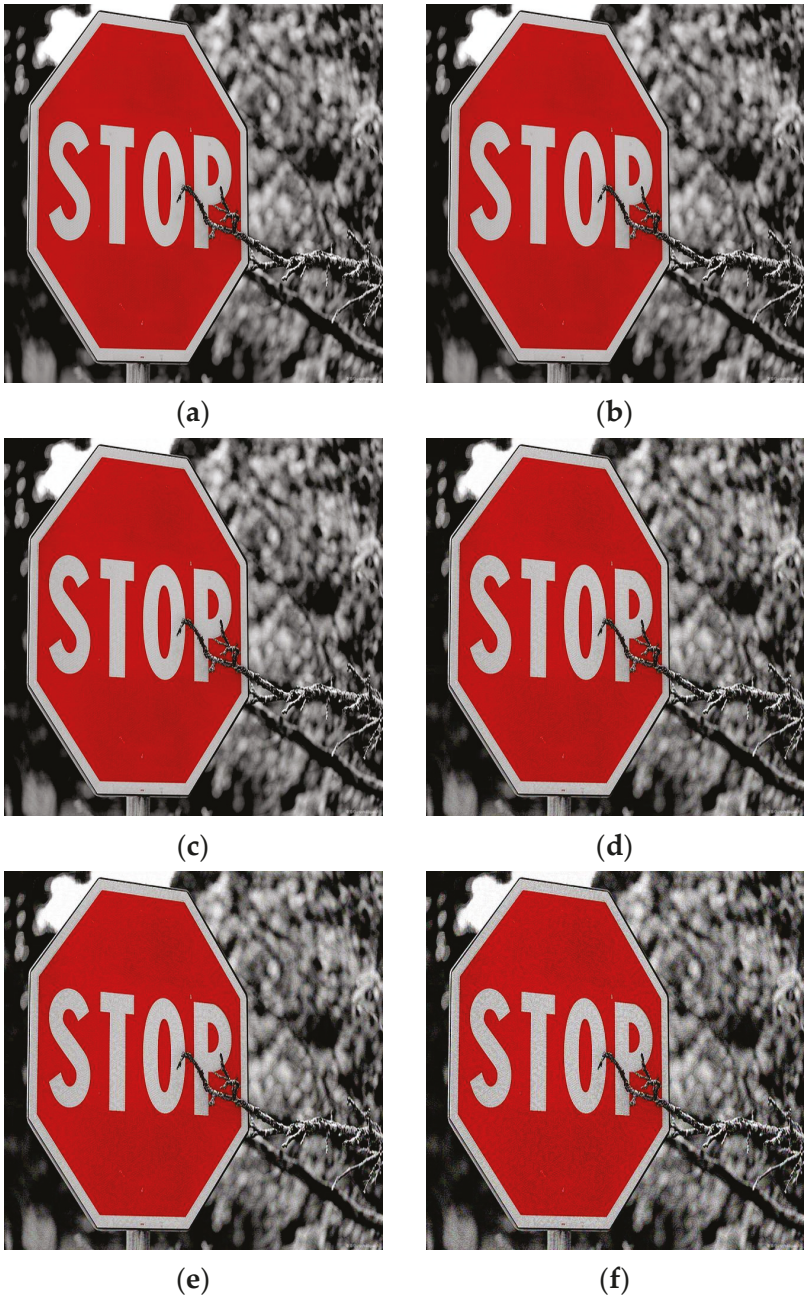


Figure 3. Images under FGSM attack. (**a**) Clean image, (**b**) $\epsilon = 0.01$, (**c**) $\epsilon = 0.03$, (**d**) $\epsilon = 0.05$, (**e**) $\epsilon = 0.07$, (**f**) $\epsilon = 0.1$.

PGD is an iterative method for generating adversarial examples that extends the idea of FGSM. Introduced by Madry et al. in 2017 [25], PGD performs multiple steps of gradient ascent to find an adversarial example, with each step projecting the perturbed input back into the valid data space, typically constrained by an ℓp-norm ball around the original input. Starting from an initial adversarial example $x'_0 = x$, PGD iteratively updates the adversarial example using the following rule:

$$x'_{t+1} = Proj_{Be(x)}\left(x'_t + a \cdot sign\left(\nabla x'_t J\left(\theta, x'_t, y\right)\right)\right) \tag{7}$$

Here, x'_t is the adversarial example at iteration t, and α is the step size. $Proj_{Be(x)}$ denotes the projection operation that ensures x' remains within an $\updownarrow p$-norm ball of radius ϵ centered around x. Figure 4 shows the clean and the attacked images with different ϵ values.

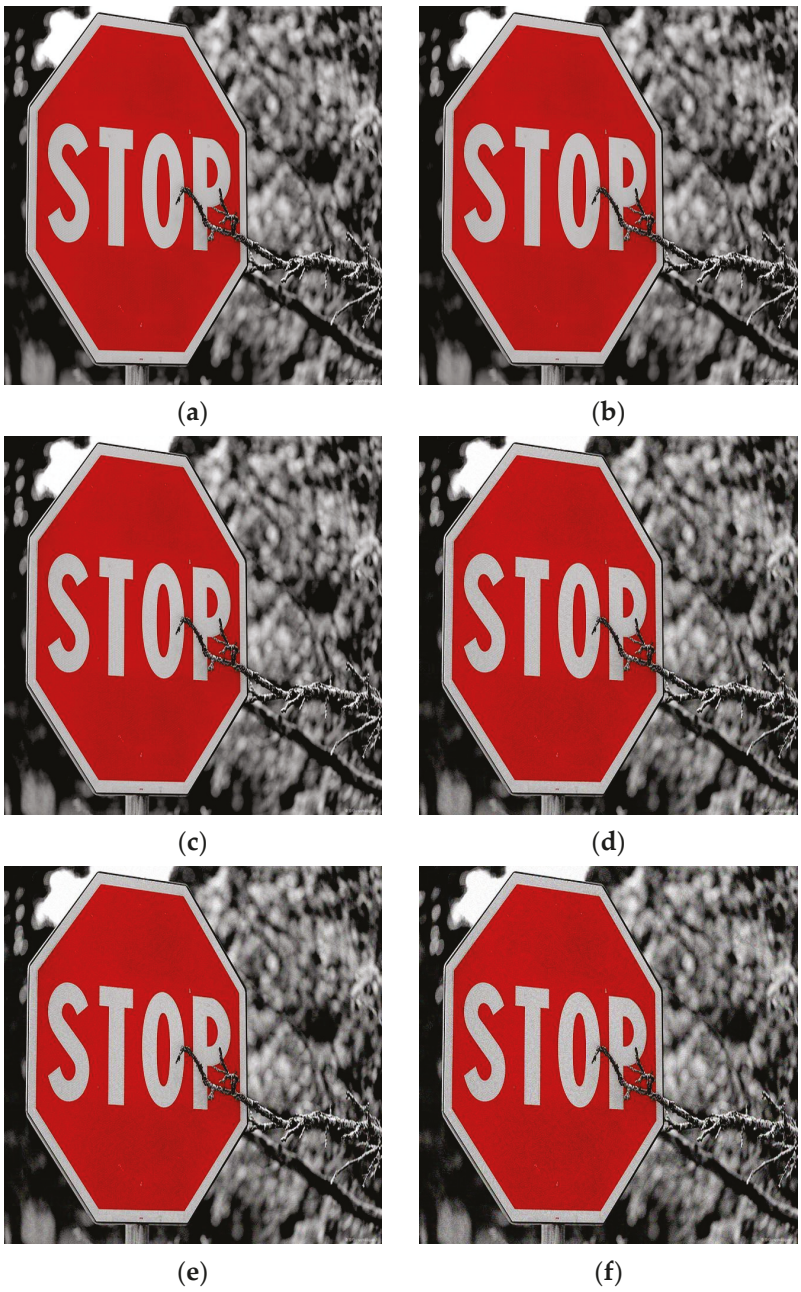


Figure 4. Images under PGD attack. (**a**) Clean image, (**b**) $\epsilon = 0.01$, (**c**) $\epsilon = 0.03$, (**d**) $\epsilon = 0.05$, (**e**) $\epsilon = 0.07$, (**f**) $\epsilon = 0.1$.

3.2.2. Image Corruptions

To thoroughly evaluate the robustness of YOLO algorithms, it is essential to test their performance under various types of image corruptions (Figure 5) that mimic real-world challenges. In this study, we introduce a diverse set of image corruptions to assess the resilience of YOLO models. These corruptions include *fog*, which simulates reduced visibility conditions; *pixelate*, which degrades image resolution; *Gaussian noise*, which adds random variations in pixel intensity; *impulse noise*, which introduces sporadic intensity spikes; *defocus blur*, which mimics the effect of an out-of-focus camera; and *motion blur*, which replicates the effect of camera or object movement. By incorporating these common yet challenging corruptions, we aim to comprehensively test and understand the robustness of YOLO models in various adverse scenarios. All the above techniques were implemented

according to a package provided by Hendrycks and Dietterich [19] in order to benchmark the robustness of neural networks.

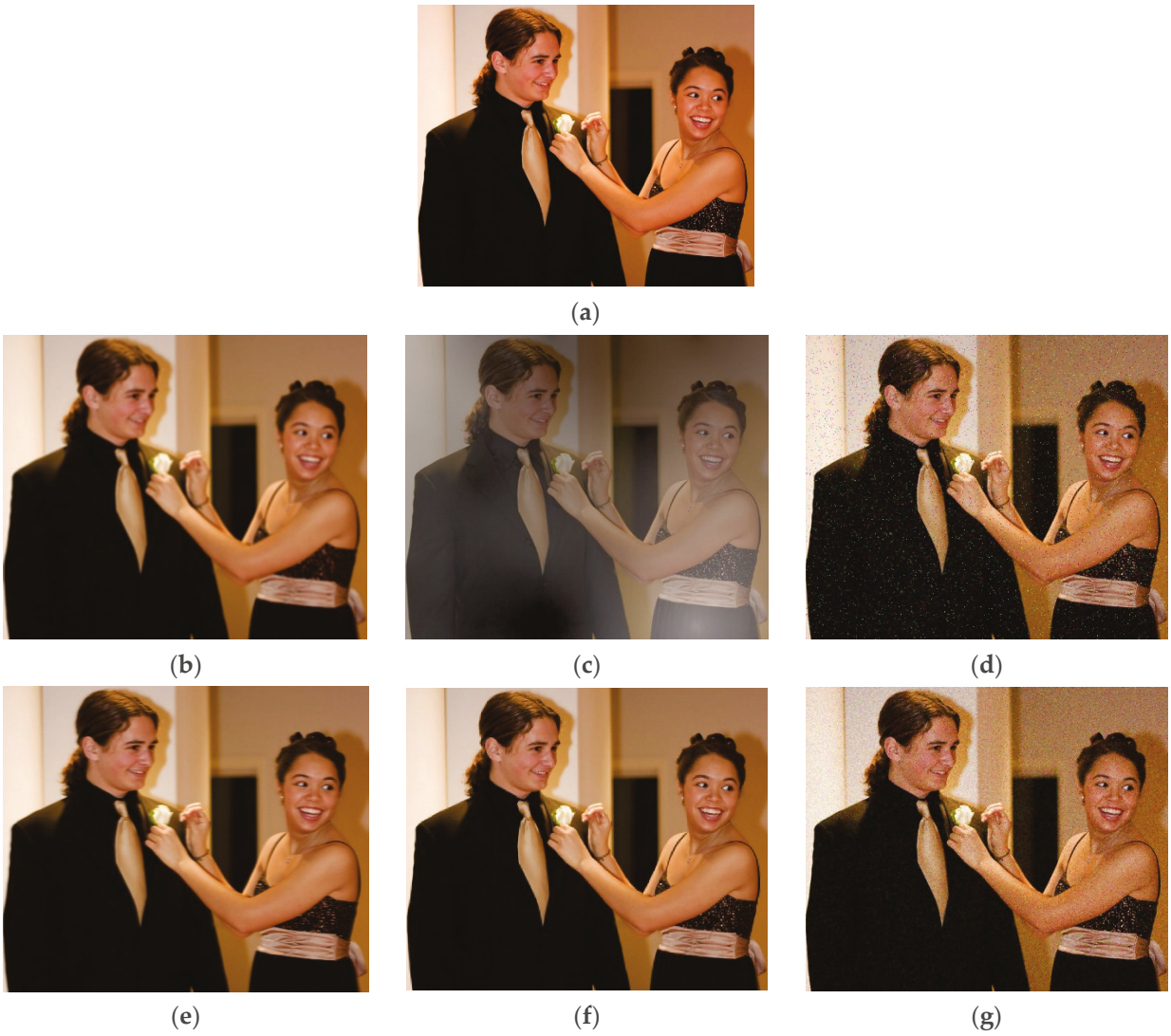


Figure 5. Example of image corruption. (**a**) Clean image, (**b**) defocus blur, (**c**) fog, (**d**) impulse noise, (**e**) motion blur, (**f**) pixelate, and (**g**) Gaussian noise.

3.3. Attack Settings and Preprocessing

We evaluated the robustness of YOLO models using 5000 test images from the COCO dataset. Each model was tested using its official implementation, ensuring that all results were based on clean, pre-trained models. Since YOLOv1 and YOLOv2 are trained and evaluated in smaller datasets, they are excluded from our experiments. Furthermore, we utilized all available sizes for each YOLO model to comprehensively evaluate their behavior in relation to model capacity. However, YOLOv3, YOLOv4, and YOLOR-CSP were tested as individual models without size variations. Additionally, we included both the official YOLOv3 and the Ultralytics implementation, referred to as YOLOv3u, to compare potential differences in robustness arising from distinct implementations.

This diverse selection allowed for a nuanced analysis of robustness across varying model capacities and implementations. From the perspective of adversarial robustness, we employed FGSM and PGD, two widely recognized and well-established attack methods that serve as strong baselines in the field. These attacks were selected due to their computational efficiency, ease of implementation, and broad acceptance in the literature as

standard tools for benchmarking robustness. FGSM provides a fast, single-step perturbation useful for stress-testing models under minimal constraints, while PGD offers a more iterative and powerful variant that simulates a stronger adversary. Together, they allow for a clear and interpretable robustness evaluation, especially when applied across a large suite of pre-trained object detection models like the YOLO series. Importantly, these attacks were implemented under a black-box threat model, which closely mirrors real-world deployment conditions. In practical scenarios—particularly when models are integrated into proprietary systems or commercial APIs—attackers typically do not have access to the model's internal parameters, architecture, or gradients. By evaluating robustness in this setting, our study aims to reflect realistic adversarial conditions and provide insights into how YOLO models perform when exposed to plausible external threats, rather than idealized white-box situations. Therefore, black-box attacks are more realistic, making the evaluation closer to actual deployment conditions. We used the Adversarial Robustness Toolbox (ART) [37] to generate adversarial images. The FGSM and PGD attacks were applied with five different perturbation levels, denoted by ϵ values of 0.01, 0.03, 0.05, 0.07, and 0.1. For the PGD attack specifically, we utilized the L∞ -norm constraint, a maximum of 10 iterations, and a step size α = 0.1. Table 1 shows the details of the adversarial attacks. Here, lower values of ϵ correspond to minimal noise, introducing subtle perturbations, while higher values of ϵ represent more significant noise levels, introducing larger perturbations. These varied ϵ levels allow for a controlled assessment of the model's robustness under increasing intensities of adversarial noise. Furthermore, we resized all images in 640×640 pixels to apply adversarial attacks. The resize led to a degradation in mAP, so we consider this mAP as a clean baseline to be fair. However, since resizing itself introduces an additional transformation, we can consider it an implicit attack.

Table 1. Configurations of adversarial attacks.

Attack Type	Norm Constraint	Perturbation Strength (ϵ)	Step Size (α)	Iterations (max_iter)
FGSM	L∞	0.01, 0.03, 0.05, 0.07, 0.1	-	1
PGD	L∞	0.01, 0.03, 0.05, 0.07, 0.1	0.1	10

For image corruption, we utilized a library developed by Hendrycks and Dietterich [19] and applied various types of noise, including *Gaussian noise*, *impulse noise*, *defocus blur*, *motion blur*, *pixelation*, and *fog noise*. These corruption types can be applied directly to images without necessitating resizing. We consider that the combination of adversarial attacks with image corruption provides a comprehensive evaluation of the robustness of YOLO models.

4. YOLO Model Series

YOLO (You Only Look Once) [10] is a real-time object detection method that frames object detection as a single regression problem rather than a series of region-based proposals. Unlike other approaches that generate multiple region proposals and classify each, YOLO directly predicts bounding boxes and class probabilities from entire images in one evaluation. This makes it highly efficient for applications requiring fast processing speeds.

YOLO divides an input image into an $S \times S$ grid, where each grid cell is responsible for predicting a fixed number of bounding boxes and their confidence scores, along with class probabilities for each box. Each prediction includes the bounding box coordinates (center x, center y, width, height), the confidence score that indicates the likelihood of the box containing an object, and the class probabilities for the detected object. The confidence score is a product of the IoU (Intersection over Union) between the predicted box and the ground truth, and the probability of the object's presence in the box.

In the YOLO framework, multiple bounding boxes are predicted for each grid cell. During training, only one bounding box predictor is assigned to be responsible for detecting a particular object. This assignment is based on which predicted box has the highest Intersection over Union (IoU) with the ground truth box. This process allows each bounding box predictor to specialize, focusing on different object sizes, aspect ratios, or classes, which helps improve the model's recall and detection accuracy.

A crucial technique used in YOLO models is Non-Maximum Suppression (NMS). NMS is a post-processing method that enhances the accuracy and efficiency of object detection by addressing the issue of multiple overlapping bounding boxes for the same object. When several predicted boxes overlap and represent the same object, NMS selects the box with the highest confidence score and removes others that are deemed redundant. This process ensures that each object is represented by a single, most accurate bounding box, improving the clarity of detection results.

4.1. YOLO v1

YOLO v1 consists of 24 convolutional layers followed by two fully connected layers. The first 20 convolutional layers were pre-trained on the ImageNet at a resolution of 224×224 pixels size. The architecture of YOLO v1 is inspired by GoogleNet [38] and Network in Network [39] which use a 1×1 convolutional layer to decrease the number of feature maps and keep the number of parameters low. Also, all layers employed a Leaky Rectified Linear Unit (LReLU), apart from the last layer that uses a linear activation function. The model was fine-tuned in PASCAL VOC 2007 and 2012 at a resolution of 448×448 pixels. The Darknet framework was used for training.

4.2. YOLO v2

The architecture of YOLOv2 [40], named darknet-19, is similar to the VGG [41] concept and consists of 19 convolutional layers and five max pooling layers. Building upon YOLOv1, YOLOv2 introduces some new advancements.

1. Batch normalization (BN) is implemented to alleviate the Internal Covariate Shift and to improve convergence.
2. High Resolution Classifier. In YOLOv2, the classifier was pre-trained in ImageNet like YOLOv1 at the resolution of 224×224 pixels. However, in the last 10 epochs of training, the resolution becomes 448×448 pixels.
3. Anchor Boxes. YOLOv2 uses anchor boxes (or prior boxes) to predict bounding boxes. These anchor boxes represent different shapes and aspect ratios, and they are used as reference templates for the objects that the model might detect. In YOLOv2, five anchor boxes were used. The width and height of the anchor boxes should match the common object of the training set. For that purpose, the authors used k-means clustering to choose the proper size of them.
4. Direct Location Prediction. Instead of predicting bounding box coordinates directly (as in YOLOv1), YOLOv2 predicts adjustments relative to these anchor boxes for each grid cell.
5. Fine Grained Features. YOLOv2 combines features from earlier and deeper layers to deal with the detection of small objects. This is carried out by using a passthrough layer that concatenates the higher-resolution features with the lower-resolution features.
6. Multi-Scale Training. Since YOLOv2 uses only convolutional and pooling layers, it can handle multiple sizes of images, making it more robust to different image sizes and to work well in a variety of input dimensions. For that reason, during training, every 10 batches, the network chooses a different image size.

4.3. YOLO v3

YOLOv3 was introduced by Redmon et al. [42] in 2018 and proposed some important changes over YOLOv2. First of all, YOLOv3 uses a bigger backbone named Darknet-53 which consists of 53 convolutional layers with 3×3 and 1×1 convolutional filters with residual connections. Also, max-pooling layers were replaced by strided convolutions. Moreover, in YOLOv3, binary cross entropy is used for classification instead of softmax. This allows for multiple label assignments in the same box. In YOLOv3, the objects are detected at three different stages/layers of the network. Assuming the input to the network is 416×416 pixels, the three feature vectors we obtain are 52×52, 26×26, and 13×13 pixels, responsible for detecting small, medium, and large objects, respectively. The network extracts features from each of the three scales using a similar concept to Feature Pyramid Networks or FPNs. Upsampling and concatenating features with different scales allow the network to learn more fine-grained information from the earlier feature maps and more meaningful semantic information from the upsampled later layer feature maps. Similarly to YOLOv2, for choosing the precise anchor boxes, the authors use k-means clustering in YOLOv3. But instead of five, in YOLOv3, nine clusters are selected, dividing them evenly across the three scales, which means three anchors for each scale per grid cell.

4.4. The Structure of Object Detectors

Henceforward, the architecture of modern object detectors is divided into three parts, the backbone, the neck, and the head, as shown in Figure 6.

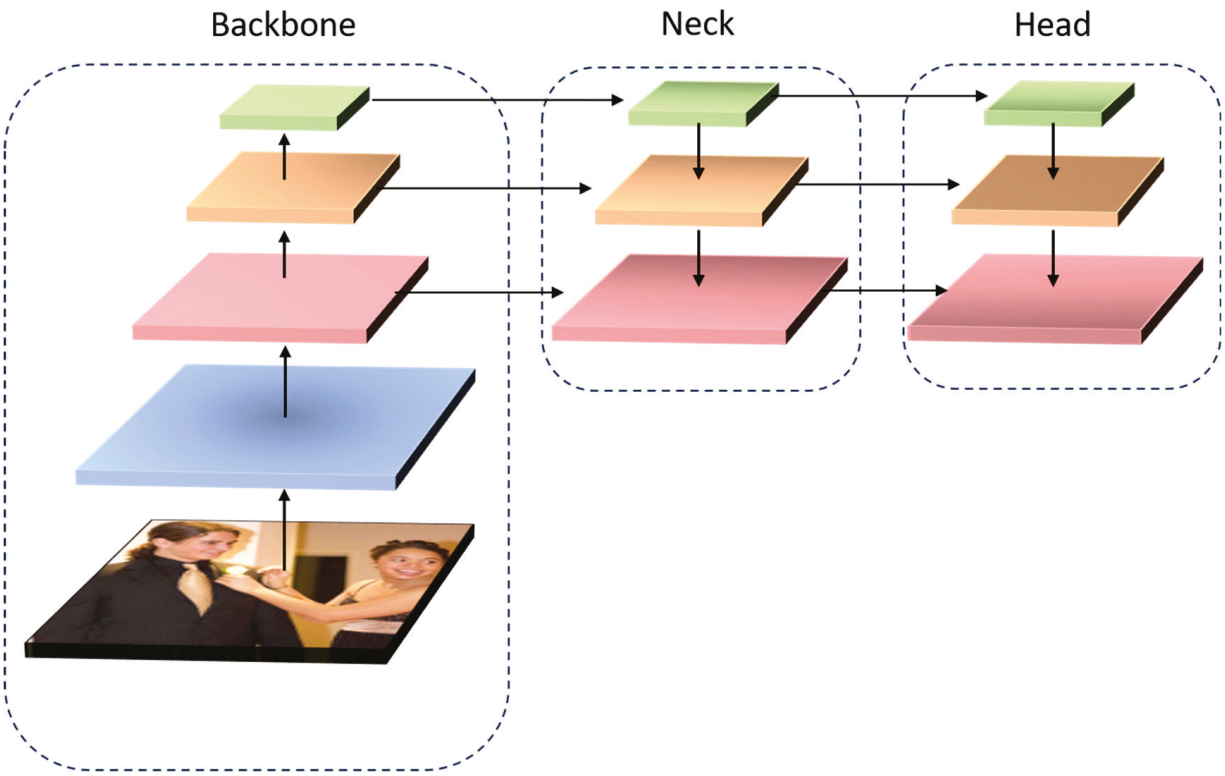


Figure 6. The structure of the modern YOLO object detectors, illustrating the three main components: the backbone for feature extraction, the neck for multi-scale feature aggregation, and the head for final bounding box prediction and classification.

The *backbone* part is responsible for extracting rich feature representations from the input image. It is typically a convolutional neural network (CNN) that is pre-trained on large-scale image classification datasets like ImageNet. Popular backbones include

ResNet [43], VGG, Darknet-53, and more. The backbone's purpose is to generate a feature map that encodes various spatial and semantic information about objects within the image.

The *neck* part further refines and processes these feature maps, often focusing on multi-scale feature aggregation. Techniques like Feature Pyramid Networks (FPNs) [44] or Path Aggregation Networks (PANet) [45] are commonly used to enhance feature maps by combining high resolution details (earlier layers) with deeper, semantically rich information (deeper layers). This multi-scale processing is crucial for detecting objects of varying sizes in an image, as it helps maintain both fine details and high-level context across different layers.

The *head* part is responsible for producing the final detection outputs from the processed feature maps. It includes layers that predict object bounding boxes, classify detected objects, and calculate objectness scores, indicating the presence of an object. The head often employs convolutional layers with regression for bounding box predictions and logistic regression for class probabilities. For models like YOLO and Faster R-CNN, the head generates bounding boxes and class scores for each anchor box at every spatial location in the feature map, allowing for the efficient detection of multiple objects.

4.5. YOLO v4

YOLOv4 was published by Bochkovskiy et al. [46] in 2020 by different authors from the previous versions. However, the main features and the philosophy remain the same. YOLOv4 introduces several new features that are divided into two main categories, Bag-of-Freebies and Bag-of-Specials. Bag-of-Freebies affect only the training procedure, increasing the training cost but without affecting inference time. On the other hand, Bag-of-Specials have an impact on inference by slightly increasing the inference cost but with an important improvement in accuracy.

The Bag-of-Specials lead to the following architecture. The authors tried several CNN architectures for the backbone like ResNeXt50 [47], EfficientNet-B3 [16], and Darknet-53. They finally chose a variation in Darknet-53 with cross-stage spatial connections (CSPNet) [48] with Mish activation function. CSP connections decrease the computation cost while preserving high accuracy. After the backbone, YOLOv4 uses spatial pyramid pooling (SPP) [49] which allows multi-scale prediction and increases the receptive fields of the network. Also, a spatial attention module (SAM) [50] block was also added. SAM assists the model in emphasizing important spatial regions. SPP and SAM along with a variation in PANet for the feature aggregation constitute the neck of YOLOv4. The original PANet algorithm adds the aggregated features, while the modified version concatenates them. As a head, YOLOv4 uses the same as YOLOv3 and anchor-based detection.

The implementation of bag-of-freebies includes several steps. Strong data augmentation techniques such as mosaic and cutmix [51] were applied along with typical techniques like random brightness, contrast, scaling, cropping, flipping, and rotation. Also, class label smoothing and DropBlock [52] were used for regularization. Furthermore, at the detection stage, CIoU [53] and cross-mini-batch normalization were used. Finally, self-adversarial training was applied to make the model robust under adversarial attacks and perturbations.

4.6. YOLO v5

YOLOv5 [54] was introduced by Glen Jocher in 2020. It is the first YOLO model that was not developed in the Darknet framework but in Pytorch. It has many similarities with YOLOv4. A modified CSPDarknet53 is used as the backbone. The authors added the Stem layer, which is a strided convolution layer with a large window size to reduce the computational cost. The architecture of the neck uses a spatial pyramid pooling fast (SPPF) layer with a modified CSP-PAN. The SPPF layer is designed to accelerate network

processing by merging features from different scales into a uniform-size feature map, enhancing computational efficiency. The head is similar to the YOLOv3 head.

Moreover, YOLOv5 uses a plethora of data augmentation techniques such as Mosaic, Copy-Paste, random affine transformations, MixUp, Albumentations, HSV, and random horizontal flip. In YOLOv5, various training strategies are implemented to enhance model performance and efficiency. Multi-scale training is utilized, where input images are randomly resized between 0.5 and 1.5 times their original size during training, enabling the model to adapt to objects of different scales. The AutoAnchor mechanism optimizes anchor boxes to better align with the characteristics of ground truth boxes in the training data, improving detection accuracy. Additionally, a warmup and cosine learning rate scheduler gradually adjusts the learning rate, helping to stabilize the training process and improve convergence. To ensure more stable model updates and reduce generalization error, an Exponential Moving Average (EMA) of model parameters was employed. Mixed precision training further contributes to faster training by performing computations in half-precision, leading to reduced memory usage. Finally, hyperparameter evolution automatically adjusts hyperparameters through an evolutionary approach, finding optimal configurations for better performance. These strategies collectively contribute to YOLOv5's efficient training and accurate detection capabilities.

4.7. YOLOR

In 2021, YOLOR (You Only Learn One Representation) was introduced by Wang et al. [55]. YOLOR emerged as one of the most advanced YOLO models, introducing a multi-task learning approach, as the model was trained to perform classification, detection, and pose estimation simultaneously. The authors combined explicit and implicit knowledge to achieve joint learning. Explicit knowledge is gained by training a model in a supervised manner for a specific task, while implicit knowledge simulates subconscious learning, similar to how humans gain latent background knowledge without direct supervision. Even though joint learning usually leads to poor feature generation, the combination of implicit and explicit knowledge presents improvements in all tasks. The authors proposed three ways to integrate implicit with explicit knowledge.

1. *Vector/Matrix/Tensor (z):* Direct use of a vector z (or higher-dimensional tensor) as implicit knowledge. Simple but effective for tasks where prior knowledge can be directly applied.
2. *Neural Network (Wz):* Transforms the implicit knowledge vector using a neural network to allow for more complex, task-specific interaction with explicit features.
3. *Matrix Factorization (ZTc):* Combines multiple vectors of implicit knowledge through factorization, enabling a rich, flexible combination of implicit knowledge with explicit learning.

4.8. YOLOX

In 2021, Zheng Ge et al. from Megvii Technology released YOLOX [56]. The authors used YOLOv3 as a baseline for Darknet53 and the SPP layer with FPN. However, they introduced five key modifications.

1. *Decoupled head:* In the previous YOLO versions, the head was responsible for the classification and the regression task for bounding boxes. However, YOLOX uses a decoupled head where two different branches of the head are responsible for classification and regression tasks independently (Figure 7).
2. *Anchor-free:* YOLOX proposes an anchor-free mechanism since it simplifies the training and decoding phase. Also, the anchor-free mechanism leads to a decreased complexity of detection heads and an increase in predictions in the image.

3. *Multi-positives:* In contrast to previous YOLO models, which selected only one positive sample per object, YOLOX assigns multiple predictions as positives, assuming that the predicted center is close to the center of the ground truth. Specifically, positive samples are those that their center fall within a 3×3 area around the ground truth center.
4. *SimOTA:* It is a key technique used in YOLOX to improve how the model assigns positive samples during training. It is based on the Optimal Transport Assignment (OTA) [57] algorithm and it aims to minimize the matching cost between predictions and ground truth objects, allowing the model to focus on the most promising predictions. By adopting SimOTA, YOLOX enhances its training efficiency, reduces the positive–negative imbalance, and ultimately improves detection accuracy.
5. *Strong augmentations:* YOLOX used the MixUP and Mosaic data augmentations and disabled them for the last 15 epochs. Also, the authors concluded that when using these augmentation techniques, the impact of pre-training is no longer as substantial.

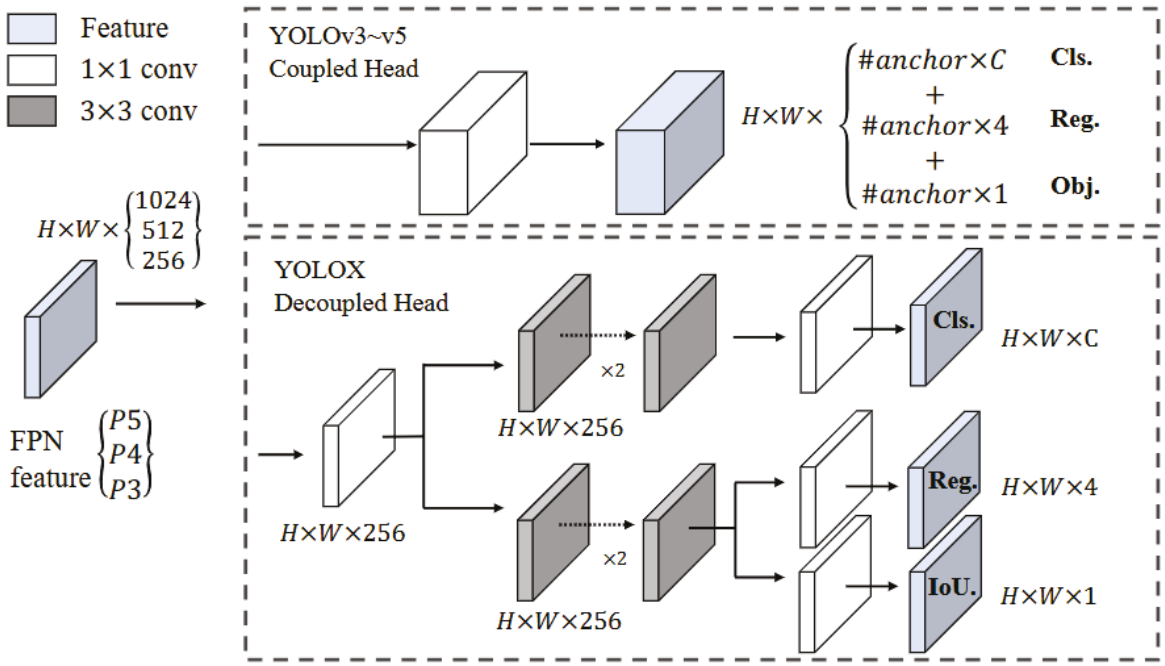


Figure 7. The head of YOLOX [43] in contrast to previous YOLO.

4.9. YOLO v6

In 2022, YOLOv6 was released by Chuyi Li et al. [58], presenting improvements in both accuracy and inference speed. The development of this new version is driven by the need to address the practical challenges often encountered in industrial applications. The advancements of YOLOv6 are summarized as follows.

- *Network Design:* The authors explored the RepVGG [59] architecture as the backbone, determining that it is well suited for small networks due to its ability to provide rich feature representations. Consequently, the smaller versions of YOLOv6 utilize RepBlock as their building block. However, in larger networks, RepBlock significantly increases computational costs, leading the authors to adopt an optimized CSP block, referred to as the CSPStackRep Block. Additionally, an enhanced PAN algorithm is employed as the neck, incorporating both RepBlocks and CSPStackRep Blocks. Finally, YOLOv6 features a decoupled head.
- *Label Assignment:* YOLOv6 employs an advanced label assignment technique similar to YOLOX, as it is also anchor-free. The authors experimented with various label

assignment methods, including SimOTA, and ultimately adopted TAL (Task-Aligned Labeling), which was introduced in TOOD [60].

- *Loss Function:* The authors experimented with several loss functions for classification, box regression, and object loss choosing VariFocal Loss [61] and SIoU [62]/GIoU [63] for classification and regression, respectively.
- *Industry-handy improvements:* YOLOv6 incorporates practical enhancements like self-distillation and extended training epochs to boost model performance. Additionally, a cosine decay strategy adjusts the balance between soft and hard labels during training, helping the student model adapt throughout different training phases.
- *Quantization and deployment:* For quantization and deployment, YOLOv6 uses RepOptimizer to produce Post-Training Quantization (PTQ)-friendly weights and combines quantization-aware training (QAT) with channel-wise distillation and graph optimization.

4.10. YOLO v7

In 2022, Wang et al. [64] published YOLO v7, which presents significant improvements in both accuracy and speed. The authors proposed several changes which are divided into architectural and trainable bag-of-freebies.

- Architecture
  1. *E-ELAN:* The authors introduced the E-ELAN block. ELAN [65] is a specific computational block designed to improve gradient flow by enabling more direct gradient paths and optimizing the way information is combined across layers. E-ELAN extends this concept, allowing for even better learning by structuring the network in a way that maintains a balance between model depth and the ability of gradients to travel through the network.
  2. *Model scaling for concatenation-based models:* Model scaling in concatenation-based models is challenging because performing depth scaling causes width scaling too, which can lead to inefficiencies in terms of computational cost and memory usage. The authors introduced a new method for scaling concatenation-based models using a uniform factor, ensuring that the scaling happens in a more balanced way, keeping the network's complexity under control while maintaining its capacity for learning.
- Trainable Bag-of-Freebies
  1. *Planned re-parameterized convolution:* Inspired by YOLOv6, YOLOv7 incorporates the RepConv architecture. However, the authors found that using this technique led to a significant decrease in the performance of models such as ResNet and DenseNet. To address this issue, they proposed RepConvN, which modifies RepConv by eliminating the identity connection, thus enhancing performance while maintaining the advantages of re-parameterized convolutions.
  2. *Auxiliary head coarse-to-fine:* In YOLOv7, the head consists of a lead and an auxiliary component. The lead head focuses on final predictions, while the auxiliary head aids in training, guided by a coarse-to-fine label assignment. This design improves both precision and recall, allowing YOLOv7 to achieve high accuracy without added inference costs.
  3. *Other trainable Bag-of-Freebies:* The authors used some other trainable tricks that they proposed. (1) Batch normalization in conv-bn-activation topology, (2) implicit knowledge, which was also used in YOLOR, and (3) Exponential Moving Average (EMA) as the final inference.

4.11. YOLO v8

In 2023, Ultralytics introduced YOLOv8 [66], which shares many similarities with YOLOv5, as both models were developed by the same company. YOLOv8 uses CSPDarknet as a backbone, like YOLOv5, but with a different module named C2f, which is a cross-stage partial bottleneck with two convolutions, replacing C3 in YOLOv5.

Another difference between these models is the fact that YOLOv8 is an anchor-free model since it does not use prior anchor boxes. Furthermore, a decoupled head was employed to process objectness score, classification, and regression independently. Finally, BCE was used for classification loss, while CIoU and DFL functions were used for bounding-box predictions. During the training process of YOLOv8, the mosaic data augmentation technique is employed, but it is deactivated during the final 10 epochs of training.

4.12. YOLO NAS

YOLO-NAS [67] is an advanced object detection architecture developed by Deci-AI using Automated Neural Architecture Construction (AutoNAC), which automates the design and optimization of deep neural networks. It delivers state-of-the-art performance in tasks like object detection by balancing speed, accuracy, and computational efficiency. Key features of YOLO-NAS include its adaptability for edge devices through hybrid quantization, which selectively reduces the precision of different parts of the neural network, ensuring smaller model size and lower power consumption without sacrificing significant accuracy. In addition to quantization, YOLO-NAS incorporates advanced machine learning techniques such as Post-Training Quantization (PTQ) and quantization-aware training (QAT). PTQ reduces the model's precision after training, enabling faster inference and lower resource consumption, while QAT trains the network to adapt to quantization effects, maintaining high accuracy even after reducing the model size. Furthermore, YOLO-NAS integrates Distribution Focal Loss (DFL), a loss function that improves the detection of objects of varying sizes, making it highly adaptable to different object detection tasks, from small-scale object recognition to large-scale scenarios.

4.13. YOLO v9

In 2024, YOLOv9 was released by Wang et al. [68], introducing two important innovations, the Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN).

PGI: Programmable Gradient Information (PGI) is introduced to mitigate the Information Bottleneck (IB) phenomenon, which often occurs in deep neural networks. PGI consists of three key components: (1) the main branch: the architecture responsible for inference; (2) the Auxiliary Reversible Branch: used only during the training phase, this branch ensures the preservation of reliable gradients through the use of reversible functions; and (3) Multi-Level Auxiliary Information: a mechanism that provides multiple levels of supervision, guiding the learning process in the main branch by supervising intermediate layers.

GELAN: Inspired by CSPNet and ELAN, the authors proposed the GELAN architecture, optimized for low complexity, fast inference, and high accuracy. GELAN enhances ELAN's ability to stack multiple convolutional blocks by enabling the stacking of any computational block. In YOLOv9, CSPNet and the planned RepConv blocks are used as computational units to further improve the model's performance.

4.14. YOLO v10

YOLOv10 was published by Wang et al. [69] and it is built on the Ultralytics package. The proposed model introduced several advancements improving both accuracy and

computational cost and can be divided into three categories: Consistent Dual Assignments for NMS-free Training, efficiency-driven model design, and accuracy-driven model design.

Consistent Dual Assignments for NMS-free Training: Previous YOLO models used one-to-many label assignment strategies, meaning that multiple predicted bounding boxes could be assigned to a single ground truth object. While this approach allows the model to learn richer features and representations, it requires the use of a post-processing method, Non-Maximum Suppression (NMS), which increases computational cost. On the other hand, one-to-one label assignment is more computationally efficient but does not provide the same level of representation learning. YOLOv10 combines these methods to leverage the benefits of both. During training, both the one-to-one and one-to-many heads are optimized, but at inference, the one-to-many head is discarded, reducing the need for NMS. To ensure optimal training, a consistent matching metric is applied to align the one-to-one head's supervision with the learning objectives of the one-to-many head.

Efficiency-Driven Model Design: YOLOv10 employed efficiency-driven changes in the head, downsampling layers and in stages with basic building blocks. The authors stated that regression error carries more impactful information for the classification task. That is why they introduced a lightweight architecture in the classification head without sacrificing significant performance. Typical YOLO architectures use convolutions with stride 2 for spatial downsampling while also doubling the number of channels. Instead, in YOLOv10, this procedure consists of two stages. Firstly, pointwise convolution is applied to change the channels and then depthwise convolution for spatial downsampling. Finally, YOLOv10 introduces a rank-guided block design to enhance computational efficiency and optimize stage-specific architecture. By conducting intrinsic rank analysis, the team identified redundancies in using uniform blocks across all stages. To address this, they developed the Compact Inverted Block (CIB), combining depthwise and pointwise convolutions for spatial and channel mixing, respectively, and integrated it into the Efficient Layer Aggregation Network (ELAN) for improved efficiency in object detection.

Accuracy-Driven model Design: YOLOv10 introduces large kernel convolution, which enhances the receptive field and as a consequence the capability of the model. Nevertheless, large kernels have a negative impact on small object detection. That is why the authors use large kernels (7×7 instead of 3×3) only in deeper stages in the CIB. They further apply structural reparameterization [70] to introduce an additional 3×3 depthwise convolution branch, helping to address optimization challenges without adding inference overhead. Furthermore, YOLOv10 enhances global feature modeling with Partial Self-Attention (PSA). Channels are split post-1×1 convolution, applying self-attention to only one subset, then recombining them with another 1×1 convolution. The query and key dimensions in MHSA are halved, and PSA is placed only after Stage 4 to reduce computational costs.

4.15. YOLO v11

YOLOv11 is the latest model at this time and it was released by Ultralytics [71] who also released YOLOv8 and YOLOv5. YOLOv11 introduces several advancements in its backbone, neck, and head parts to enhance object detection efficiency and accuracy. The backbone employs a new C3k2 block, replacing the older C2f block from YOLOv8, which optimizes feature extraction with two smaller convolutions, improving computational efficiency. The SPPF block is retained, and a new Cross Stage Partial with Spatial Attention (C2PSA) block is added to enhance spatial attention, allowing the model to focus on important regions. In the neck, C3k2 improves feature aggregation, further enhancing speed and processing efficiency. The head utilizes multiple C3k2 and CBS (Convolution-BatchNorm-Silu) blocks, with C3k2 modules adaptable for different convolutional sizes to refine multi-scale features. The final detect layer produces bounding boxes

and class scores, leveraging these innovations for faster and more precise object localization and classification.

Figure 8 represents the evolution of YOLO object detection architectures, from YOLOv3 to YOLOv11. Each block represents a specific YOLO version, highlighting its backbone network, architectural improvements, and key innovations.

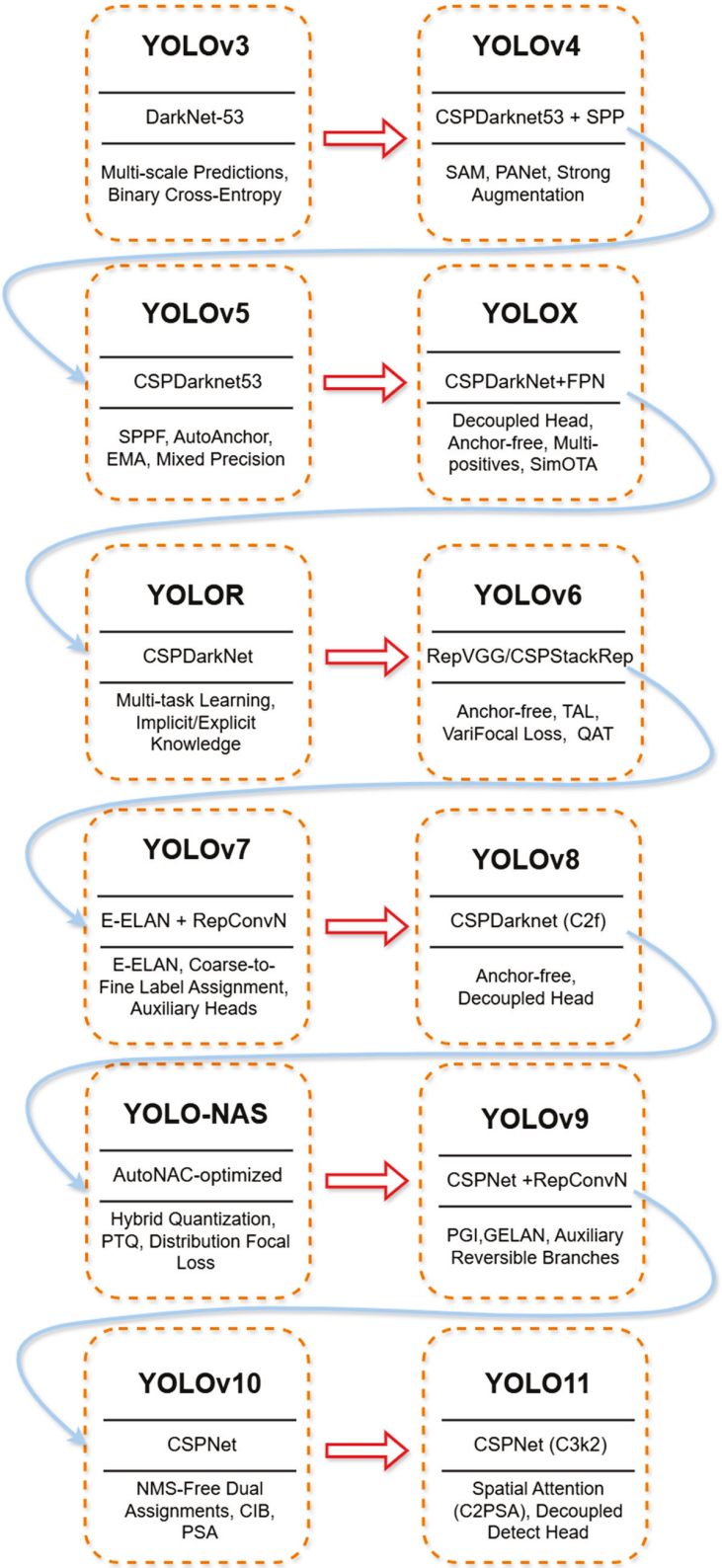


Figure 8. Evolution of YOLO object detection architectures.

5. Experimental Study in Robustness

Evaluating the robustness of YOLO object detection models under adversarial attacks and corrupted images is crucial for their deployment in real-world applications, particularly in safety-critical systems like autonomous vehicles and surveillance. Adversarial attacks, which involve adding subtle perturbations to images, can mislead YOLO models into misclassifying or failing to detect objects, posing severe safety risks. Similarly, common image corruption types—such as blurring, noise, or variations in brightness—can degrade model performance, affecting detection accuracy and reliability. Robustness evaluation helps identify the vulnerabilities of YOLO models under these conditions and drives the development of defense mechanisms to enhance their resilience. While teams in YOLO development focus on improvement in high accuracy and low computational cost, the robustness evaluation is essential to ensuring that YOLO models maintain high accuracy and reliability across various conditions, ultimately making them safer and more reliable for deployment in unpredictable environments.

Results

In this section, we present the evaluation results of the YOLO models under the previously described attacks. Table 2 provides the nominal performance metrics for each model, including their parameters. In subsequent tables, we document the degradation in accuracy observed under each attack relative to the baseline (clean) accuracy.

Table 2. YOLO models with their nominal mAP (%) and corresponding parameters.

Model	mAP	Parameters (M)	Model	mAP	Parameters (M)
YOLOv3	28.70	65.30	YOLOv8mu	50.10	25.90
YOLOv3u	51.70	103.80	YOLOv8lu	52.90	43.70
YOLOv4	28.20	63.90	YOLOv8xu	54.00	68.20
YOLOv5nu	34.30	2.70	YOLOv9-t	46.80	2.00
YOLOv5su	43.00	9.20	YOLOv9-s	51.40	7.20
YOLOv5mu	49.00	25.10	YOLOv9-m	53.00	20.10
YOLOv5lu	52.20	53.20	YOLOv9-c	53.00	25.50
YOLOv5xu	53.20	97.30	YOLOv9-e	55.60	58.10
YOLOX-s	40.50	9.00	YOLO NAS S	46.50	19.00
YOLOX-m	46.90	25.30	YOLO NAS M	50.40	51.10
YOLOX-l	49.70	54.20	YOLO NAS L	51.30	66.90
YOLOX-x	51.10	99.10	YOLOv10n	38.50	2.30
YOLOR csp	51.10	99.80	YOLOv10s	46.30	7.20
YOLOv6-N	37.50	4.65	YOLOv10m	51.10	15.40
YOLOv6-S	45.10	18.54	YOLOv10l	53.20	24.40
YOLOv6-M	50.00	34.86	YOLOv10x	54.40	29.50
YOLOv6-L	52.80	59.61	YOLOv11n	39.50	2.60
YOLOv7	51.20	36.90	YOLOv11s	47.00	9.40
YOLOv7x	53.10	71.30	YOLOv11m	51.50	20.10
YOLOv8nu	37.10	3.20	YOLOv11l	53.40	25.30
YOLOv8su	44.70	11.20	YOLOv11x	54.70	56.90

Tables 3 and 4 present the results from the FGSM and PGD attacks, respectively. The mAP column reports the nominal performance metrics of each YOLO model on the COCO dataset, while the clean mAP column shows the models' performance on resized, unaltered images. The remaining columns represent the degradation in mAP relative to the clean mAP for each corresponding epsilon value.

Table 3. YOLO model's performance under FGSM attack (bold numbers represent the best performance in each case).

Model	mAP	Clean mAP	ϵ = 0.01	ϵ = 0.03	ϵ = 0.05	ϵ = 0.07	ϵ = 0.1
YOLOv3	28.70	27.80	0.10	0.70	2.40	3.80	6.40
YOLOv3u	51.70	47.90	0.20	1.20	3.30	5.20	8.90
YOLOv4	28.20	28.90	0.20	0.80	2.40	3.70	6.20
YOLOv5nu	34.30	31.10	0.20	1.40	4.30	6.70	11.10
YOLOv5su	43.00	39.20	0.10	1.40	4.00	6.50	11.50
YOLOv5mu	49.00	45.60	0.30	1.10	3.60	5.90	11.00
YOLOv5lu	52.20	48.60	**0.00**	0.90	3.10	5.00	9.00
YOLOv5xu	53.20	49.90	0.20	0.90	2.90	4.60	8.60
YOLOX-S	40.50	27.10	0.10	0.70	2.30	3.30	5.60
YOLOX-M	46.90	32.10	0.10	**0.60**	1.90	3.00	5.00
YOLOX-L	49.70	34.40	0.20	0.80	1.90	3.10	5.00
YOLOX-X	51.10	35.40	0.20	**0.60**	**1.60**	**2.40**	**4.30**
YOLOR	51.10	47.30	0.20	1.00	2.90	4.70	8.30
YOLOv6-N	37.50	35.30	0.20	0.90	3.50	5.40	9.40
YOLOv6-S	45.10	41.80	0.20	1.10	3.50	5.40	9.30
YOLOv6-M	50.00	46.50	0.20	1.00	3.00	4.60	8.20
YOLOv6-L	52.80	49.70	0.20	0.90	2.70	4.20	7.60
YOLOv7	51.20	47.20	0.10	1.10	3.30	5.10	9.30
YOLOv7x	53.10	49.50	0.20	1.10	3.30	4.90	7.60
YOLOv8nu	37.10	33.80	0.30	1.40	4.00	6.40	11.30
YOLOv8su	44.70	41.40	0.20	1.50	4.00	6.20	10.80
YOLOv8mu	50.10	46.90	0.20	1.00	3.10	4.80	8.80
YOLOv8lu	52.90	49.70	**0.00**	0.90	2.70	4.40	7.80
YOLOv8xu	54.00	50.70	0.20	0.90	2.60	4.00	7.20
YOLOv9-t	38.30	35.40	0.10	1.40	4.60	6.90	11.00
YOLOv9-s	46.80	43.70	0.20	1.30	3.90	5.70	9.90
YOLOv9-m	51.40	47.80	0.20	1.00	3.00	4.80	8.40
YOLOv9-c	53.00	49.70	0.20	1.10	3.10	5.10	8.90
YOLOv9-e	**55.60**	52.50	0.20	0.70	2.30	3.80	7.00
YOLO NAS S	46.50	42.80	0.30	1.20	3.50	5.40	9.50
YOLO NAS M	50.40	46.90	0.10	0.90	3.10	4.70	8.00
YOLO NAS L	51.30	47.50	0.30	0.90	2.60	4.10	7.60
YOLOv10n	38.50	36.10	0.10	1.30	4.30	6.40	10.80
YOLOv10s	46.30	43.50	0.20	1.30	3.80	5.90	10.20
YOLOv10m	51.10	47.90	0.20	1.00	3.10	5.00	8.90
YOLOv10l	53.20	50.00	0.10	0.80	2.70	4.30	8.00
YOLOv10x	54.40	51.00	0.20	0.80	2.60	4.00	7.50
YOLOv11n	39.50	36.80	0.30	1.40	4.50	6.80	12.10
YOLOv11s	47.00	43.80	0.30	1.40	4.10	6.20	10.50
YOLOv11m	51.50	47.70	0.20	1.10	3.20	5.00	9.00
YOLOv11l	53.40	50.00	0.20	1.00	3.00	4.50	7.90
YOLOv11x	54.70	51.60	0.20	**0.60**	2.20	3.60	6.50

YOLOX-X and larger YOLO versions consistently showed lower accuracy degradation across various perturbation levels. The superior robustness of YOLOX models can be attributed to their anchor-free design and decoupled heads. The anchor-free structure inherently reduces complexity, minimizing the potential vulnerabilities associated with anchor box predictions. Furthermore, the decoupled head separates classification from bounding box regression, limiting adversarial impact because perturbations typically exploit coupled predictions.

Table 4. YOLO model's performance under PGD attack (bold numbers represent the best performance in each case).

Model	mAP	Clean mAP	ϵ = 0.01	ϵ = 0.03	ϵ = 0.05	ϵ = 0.07	ϵ = 0.1
YOLOv3	28.70	27.80	0.10	0.50	1.70	2.20	3.90
YOLOv3u	51.70	47.90	0.20	0.80	2.60	3.20	6.20
YOLOv4	28.20	28.90	0.20	0.80	2.00	2.60	4.60
YOLOv5nu	34.30	31.10	0.20	1.00	3.70	4.60	8.00
YOLOv5su	43.00	39.20	0.20	0.80	3.20	4.30	8.20
YOLOv5mu	49.00	45.60	0.30	0.80	2.80	3.60	7.40
YOLOv5lu	52.20	48.60	**0.00**	0.50	2.30	4.10	5.90
YOLOv5xu	53.20	49.90	0.10	0.70	2.30	2.90	5.50
YOLOX-S	40.50	27.10	0.10	0.40	1.70	2.30	3.80
YOLOX-M	46.90	32.10	0.20	0.40	1.50	1.80	3.30
YOLOX-l	49.70	34.40	0.20	0.60	1.50	1.90	3.40
YOLOX-X	51.10	35.40	0.10	0.40	**1.30**	**1.50**	**2.80**
YOLOR	51.10	47.30	0.20	0.70	2.30	3.00	5.70
YOLOv6-N	37.50	35.30	0.20	0.90	2.80	3.80	6.30
YOLOv6-S	45.10	41.80	0.10	0.70	2.70	3.60	6.50
YOLOv6-M	50.00	46.50	0.20	0.70	2.40	2.80	5.40
YOLOv6-L	52.80	49.70	0.20	0.50	2.10	2.80	5.20
YOLOv7	51.20	47.20	0.10	0.70	2.40	3.10	6.10
YOLOv7x	53.10	49.50	0.20	0.70	2.40	2.90	5.80
YOLOv8nu	37.10	33.80	0.30	1.00	3.50	4.30	7.80
YOLOv8su	44.70	41.40	0.10	0.90	3.00	4.00	7.20
YOLOv8mu	50.10	46.90	0.10	0.60	2.40	3.20	6.00
YOLOv8lu	52.90	49.70	0.10	0.70	2.20	2.70	5.20
YOLOv8xu	54.00	50.70	0.20	0.70	2.10	2.60	4.90
YOLOv9-t	38.30	35.40	0.20	1.10	3.70	4.60	8.00
YOLOv9-s	46.80	43.70	0.20	0.90	3.10	3.90	6.60
YOLOv9-m	51.40	47.80	0.20	0.60	2.40	3.10	5.60
YOLOv9-c	53.00	49.70	0.20	0.70	2.60	3.20	5.80
YOLOv9-e	**55.60**	52.50	0.10	0.50	2.00	2.60	4.50
YOLO NAS S	46.50	42.80	0.30	0.80	2.80	3.20	6.30
YOLO NAS M	50.40	46.90	0.10	0.60	2.50	3.00	5.40
YOLO NAS L	51.30	47.50	0.40	0.70	2.00	2.40	4.80
YOLOv10n	38.50	36.10	0.20	1.00	3.50	4.60	7.40
YOLOv10s	46.30	43.50	0.20	1.00	3.30	4.20	7.10
YOLOv10m	51.10	47.90	0.20	0.70	2.50	3.20	6.00
YOLOv10l	53.20	50.00	0.20	0.60	2.10	2.60	5.30
YOLOv10x	54.40	51.00	0.10	0.50	1.80	2.40	4.80
YOLOv11n	39.50	36.80	0.20	1.10	3.50	4.50	8.20
YOLOv11s	47.00	43.80	0.20	1.00	3.40	4.20	7.50
YOLOv11m	51.50	47.70	0.20	0.80	2.50	3.20	5.90
YOLOv11l	53.40	50.00	0.10	0.70	2.40	2.70	5.40
YOLOv11x	54.70	51.60	0.10	**0.10**	1.60	2.10	4.00

YOLOX-X again demonstrated superior resilience compared to other models, notably due to its SimOTA positive sample assignment strategy, which optimizes the training by focusing on predictions closest to the ground truth. This selective optimization reduces susceptibility to gradient-based adversarial perturbations, such as those produced by PGD attacks. Larger variants like YOLOv11-X and YOLOv9-E also exhibited robustness due to deeper architectures and improved feature aggregation, facilitating more stable internal representations even when perturbed.

Table 5 shows the degradation models experienced under image corruptions such as fog, pixelate, gaussian noise, impulse noise, defocus blur, motion blur, and adversarial attacks from FGSM and PGD with ϵ = 0.01.

Table 5. Degradation of YOLO models in image corruptions and adversarial attacks with $\epsilon = 0.1$ (bold numbers represent the best performance in each case).

Model	Clean	Fog	Pixelate	GN	IN	DB	MB	FGSM	PGD
YOLOv3	28.70	1.30	**0.70**	**4.40**	11.80	5.50	4.30	6.40	3.90
YOLOv3u	51.70	1.40	9.20	7.50	12.20	6.70	7.80	8.90	6.20
YOLOv4	28.20	1.70	7.70	4.70	**8.90**	**3.70**	**4.20**	6.20	4.60
YOLOv5nu	34.30	3.80	3.40	10.30	12.90	5.40	7.30	11.10	8.00
YOLOv5su	43.00	3.60	8.10	9.70	15.60	6.60	8.60	11.50	8.20
YOLOv5mu	49.00	1.70	8.40	8.00	16.40	6.80	8.40	11.00	7.40
YOLOv5lu	52.20	1.30	7.30	7.30	14.30	6.60	8.00	9.00	5.90
YOLOv5xu	53.20	1.20	7.90	6.70	13.50	6.20	8.00	8.60	5.50
YOLOX-s	40.50	1.50	6.70	7.00	10.40	6.50	6.60	5.60	3.80
YOLOX-m	46.90	1.30	7.10	6.40	10.50	6.70	7.00	5.00	3.30
YOLOX-l	49.70	**0.80**	6.80	5.80	10.40	6.90	7.00	5.00	3.40
YOLOX-x	51.10	0.90	6.30	5.00	**8.90**	6.70	6.90	**4.30**	**2.80**
YOLOR csp	51.10	1.20	10.40	8.00	13.30	7.80	8.10	8.30	5.70
YOLOv6-N	37.50	2.80	3.30	7.80	10.00	5.70	6.30	9.40	6.30
YOLOv6-S	45.10	3.20	4.90	8.30	12.70	7.60	7.50	9.30	6.50
YOLOv6-M	50.00	1.70	9.20	7.30	13.40	8.30	8.60	8.20	5.40
YOLOv6-L	52.80	1.20	8.00	6.60	11.60	7.90	7.70	7.60	5.20
YOLOv7	51.20	1.30	11.70	7.20	13.90	8.30	8.70	9.30	6.10
YOLOv7x	53.10	1.20	12.00	7.20	13.50	8.70	8.90	7.60	5.80
YOLOv8nu	37.10	3.40	5.10	9.90	14.20	5.50	7.40	11.30	7.80
YOLOv8su	44.70	3.50	9.10	9.40	15.70	6.50	8.10	10.80	7.20
YOLOv8mu	50.10	1.10	9.20	7.20	15.10	6.30	7.40	8.80	6.00
YOLOv8lu	52.90	1.30	8.10	6.50	12.40	6.10	7.80	7.80	5.20
YOLOv8xu	54.00	1.20	9.30	6.60	12.40	6.10	7.50	7.20	4.90
YOLOv9-t	38.30	3.60	4.80	9.30	13.50	6.70	7.80	11.00	8.00
YOLOv9-s	46.80	3.70	7.70	9.00	14.80	7.60	8.50	9.90	6.60
YOLOv9-m	51.40	1.30	8.40	7.60	16.40	7.80	8.80	8.40	5.60
YOLOv9-c	53.00	1.70	9.90	7.40	15.70	6.60	8.60	8.90	5.80
YOLOv9-e	**55.60**	1.10	7.80	5.80	11.70	6.20	8.00	7.00	4.50
YOLO NAS S	46.50	2.40	9.50	8.20	14.00	7.50	8.20	9.50	6.30
YOLONAS M	50.40	1.40	8.80	6.60	13.90	7.90	8.20	8.00	5.40
YOLO NAS L	51.30	1.50	7.60	6.60	13.60	8.10	8.20	7.60	4.80
YOLOv10n	38.50	3.80	4.80	9.20	12.80	6.40	6.90	10.80	7.40
YOLOv10s	46.30	3.60	6.40	9.30	14.60	7.90	8.10	10.20	7.10
YOLOv10m	51.10	1.80	9.10	8.00	13.90	8.30	7.90	8.90	6.00
YOLOv10l	53.20	1.70	7.90	6.70	12.30	8.40	8.10	8.00	5.30
YOLOv10x	54.40	1.70	9.90	6.80	13.20	8.70	8.50	7.50	4.80
YOLOv11n	39.50	4.00	5.60	10.40	15.00	6.10	7.80	12.10	8.20
YOLOv11s	47.00	2.30	10.80	9.10	16.90	6.40	8.70	10.50	7.50
YOLOv11m	51.50	1.50	12.00	7.30	16.80	7.00	8.80	9.00	5.90
YOLOv11l	53.40	1.60	12.00	7.10	16.60	6.90	8.80	7.90	5.40
YOLOv11x	*54.70*	1.20	10.30	6.20	14.30	6.60	8.80	6.50	4.00

YOLOv3 and YOLOv4 outperformed newer models in specific corruptions like pixelation and Gaussian noise due to their comparatively simpler feature extraction pipelines, which avoid excessive sensitivity to fine-grained perturbations. Conversely, YOLOX performed strongly against fog and blur effects because of its optimized multi-scale feature aggregation, enabling better adaptation to texture and spatial distortions. Models with explicit data augmentation strategies such as YOLOv7 and YOLOv8 maintained decent robustness but exhibited specific weaknesses to pixelation noise, indicating a trade-off between robustness to naturally occurring distortions versus artificial adversarial perturbations.

Conversely, YOLO models relying heavily on anchor boxes, particularly YOLOv5 and earlier variants, show greater sensitivity to adversarial attacks, as perturbations often significantly disrupt anchor-dependent predictions. Older architectures like YOLOv3, despite lacking modern efficiency features, display surprising robustness in specific corruptions due to their straightforward structure, which inadvertently limits the exploitation of complex model features by adversarial attacks. Models employing heavy data augmentation (e.g., YOLOv8, YOLOv11) showed robustness to environmental distortions but vulnerability to pixel-level adversarial perturbations, highlighting the nuanced trade-offs in model training strategies.

Figure 9 is a heatmap that visualizes the average performance degradation (AVG Degradation) of various YOLO models under adversarial attacks or image corruptions, with lighter colors indicating higher robustness. The models are ordered by degradation level, facilitating a clear comparison of their relative resilience.

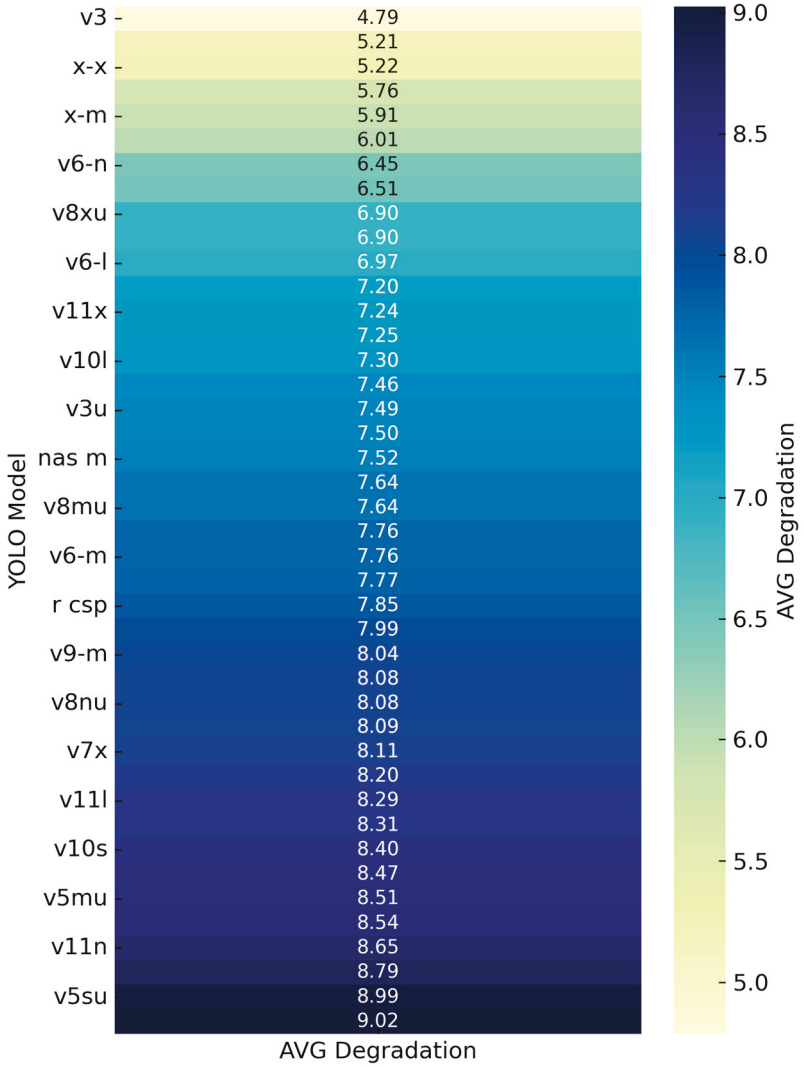


Figure 9. Heatmap with average degradation across all attacks.

To gain a more nuanced understanding of model robustness beyond overall mAP degradation, we conducted an area-wise breakdown based on object sizes for the most recent YOLO models and their largest version, including YOLOX, which also demonstrated the highest overall robustness. Following COCO standards, objects were categorized into small (area $< 32^2$ pixels), medium ($32^2 \leq$ area $< 96^2$ pixels), and large (area $\geq 96^2$ pixels). Across most YOLO variants, the small object category consistently exhibited the greatest performance degradation under FGSM attacks, which aligns with the intuition that small objects are more susceptible to slight pixel-level perturbations. For instance, YOLOX, YOLOv8, and YOLOv9 all showed their highest FGSM-induced mAP drop in the small object subset. Interestingly, under PGD, a more iterative and powerful attack, medium-sized objects experienced the greatest degradation in YOLOX, YOLOv8, YOLOv10, and YOLOv11. YOLOv9 was an exception, with small objects remaining the most vulnerable even under PGD. Notably, YOLOv10 showed an unusual pattern, with large objects being the most affected under FGSM. This divergence may indicate model-specific weaknesses

in handling scale-invariant features under adversarial noise. The summarized results of area-based vulnerability across models and attack types are provided in Table 6.

Table 6. Most affected object size category by attack type across YOLO models.

Model	FGSM (Most Affected Area)	PGD (Most Affected Area)
YOLOX	Small	Medium
YOLOv8	Small	Medium
YOLOv9	Small	Small
YOLOv10	Large	Medium
YOLO11	Medium	Medium

A class-level analysis further reveals how adversarial attacks affect semantic categories differently. The results highlight that some classes remain remarkably stable even under severe attacks, while others experience substantial degradation. For instance, YOLOX and YOLOv8 identified "toothbrush" as the most robust class under FGSM, while YOLOv9, YOLOv10, and YOLOv11 found "toaster" among the most robust across both FGSM and PGD, even outperforming their clean performance in YOLOv9. This could be attributed to their distinct shapes or relatively less visual ambiguity. In contrast, classes like "bear", "microwave", "suitcase", and "hair drier" were among the most vulnerable, showing substantial mAP degradation. YOLOv11 notably identified "suitcase" as the most degraded under both FGSM and PGD, indicating potential sensitivity to structured textures or occlusions. These findings suggest that robustness is not uniform across semantic categories and that certain classes could act as stress tests when evaluating model vulnerability. They also point to opportunities for targeted data augmentation or architecture refinement to improve resilience in weaker categories. An overview of the most robust and vulnerable classes for each model and attack type is presented in Table 7.

Table 7. Most robust and most vulnerable object classes under FGSM and PGD attacks.

Model	FGSM Robust	PGD Robust	FGSM Vulnerable	PGD Vulnerable
YOLOX	Toothbrush	Toothbrush	Bear	Hair drier
YOLOv8	Toothbrush	Kite	Suitcase	Toaster
YOLOv9	Toaster	Toaster	Mouse	Hot dog
YOLOv10	Kite	Kite	Microwave	Toaster
YOLO11	Toaster	Toaster	Suitcase	Suitcase

6. Discussion

Robustness in object detectors is crucial since they are used in critical domains such as autonomous driving, surveillance, etc. This study highlights the relative robustness of different YOLO model versions when exposed to adversarial attacks, such as FGSM, PGD, and image corruptions like fog, pixelate, gaussian noise, impulse noise, defocus blur, and motion blur. This section discusses the results of the conducted simulations and analyzes the behavior of the models under attack.

Specifically, under the FGSM attack, the X-Large YOLOX model demonstrated the highest robustness. While the large versions of YOLOv5 and YOLOv8 maintained their accuracy under attack with an epsilon value of 0.01, they experienced greater degradation than YOLOX at higher epsilon values. Following YOLOX, the YOLOv4, YOLOv3, and YOLO 11x models, demonstrated a notable degree of robustness. A similar pattern is observed under PGD attacks. The large YOLOv5 model maintains its accuracy at an epsilon

value of 0.01; however, YOLOX models exhibit lower degradation across the remaining epsilon values. Following YOLOX, YOLOv3, YOLO 11x, YOLOv9-e, and YOLOv4 present a good performance in that order. Furthermore, results indicate that all sizes of YOLOX models exhibit superior robustness compared to other models.

When images are attacked by applying some types of image corruption, the results are slightly different. Firstly, each corruption method introduces a distinct type of noise, which explains why a model that demonstrates robustness against one type of noise may not exhibit the same level of resilience against others. However, the top-performing models across each type of corruption are consistently YOLOX, YOLOv3, and YOLOv4. More precisely, the results for each corruption are as follows.

Fog: The introduction of fog noise causes the smallest degradation in performance among all corruptions. Large YOLOX achieved the best performance, losing only 0.8 mAP. However, a plethora of other models such as YOLOX-x, YOLOv9-e, YOLOv8m, YOLOv7x, and others have a similar behavior of maintaining high mAP. On the other hand, YOLO11n, YOLOv9-s, and YOLOv5n presented the worst performance.

Pixelate: In pixelate corruption, YOLOv3 surprisingly loses only 0.7% mAP while the other models have a significantly more significant decrease in their performance. YOLOv5n and YOLOv6n were the second-best models with 3.3% and 3.4% degradation, respectively. YOLOv7, YOLOR-CSP, and all sizes of YOLO11 except for nano achieved the worst performance.

Gaussian Noise (GN): In Gaussian noise, YOLOv3 is again the most robust model. Nevertheless, near to its performance are YOLOv4 and YOLOX-x. YOLO5n, YOLOv8n, and YOLO11n were the least robust models for this type of noise.

Impulse Noise (IN): YOLOv4 and YOLOX-x were the most robust models under attack with impulse noise as they were degraded by 8.9% mAP. Impulse noise was the strongest type of noise causing the biggest degradation in models. YOLO11, YOLOv5, YOLOv8, and YOLOv9 present some of the worst results.

Defocus Blur (DB): The defocus blur degradation scores range from 3.7% to around 8.7%. YOLOv4 has the best performance under defocus blur with a score of 3.7%, indicating it is the most robust model against this specific corruption type. Other models with relatively high robustness for this type of noise include YOLOv3 (5.5%) and YOLOX-s (6.5%), which suggests that these models are also more resilient to defocus blur, albeit with slightly higher degradation than YOLOv4. Models such as YOLOv9-m (7.8%) and YOLOv11m (7.9%) show higher degradation, indicating they are more susceptible to performance drops under defocus blur.

Motion Blur (MB): YOLOv4 achieved the best performance in MB noise. YOLOv3 follows with very good performance too. YOLOX models also perform relatively well with this type of noise. In contrast, all sizes of YOLOv7, YOLOv9, and YOLO11 seem to be more susceptible in MB.

In Figure 10, the overall behavior of the models is visualized. As is shown, the blue frame contains the models with high clean mAP performance while the red frame includes the models with the least mAP degradation. The intersection of these two frames provides us with models that achieve high mAP performance in both clean and attacked images. The size of the X marker in the figure is proportional to the number of parameters of each model; larger markers indicate models with higher parameter counts. It is observed that YOLOX-x and YOLOX-l are the only models that achieve both goals.

YOLOX models (particularly YOLOX-l and YOLOX-x) consistently exhibit low degradation across a range of attacks, such as FGSM and PGD, as well as various corruption types like Gaussian noise (GN) and pixelation. This suggests a high level of robustness in the YOLOX architecture, especially for large versions, making them suitable for environments

where adversarial robustness is critical. YOLOX and certain versions of YOLOv3, YOLOv4, and YOLOv9-e are among the top-performing models across multiple corruption types. For example, they perform well under Gaussian noise (GN), defocus blur (DB), and motion blur (MB). This consistent robustness across diverse noise types implies that these models are better at handling various types of data imperfections, a key feature for applications in dynamic and uncontrolled environments.

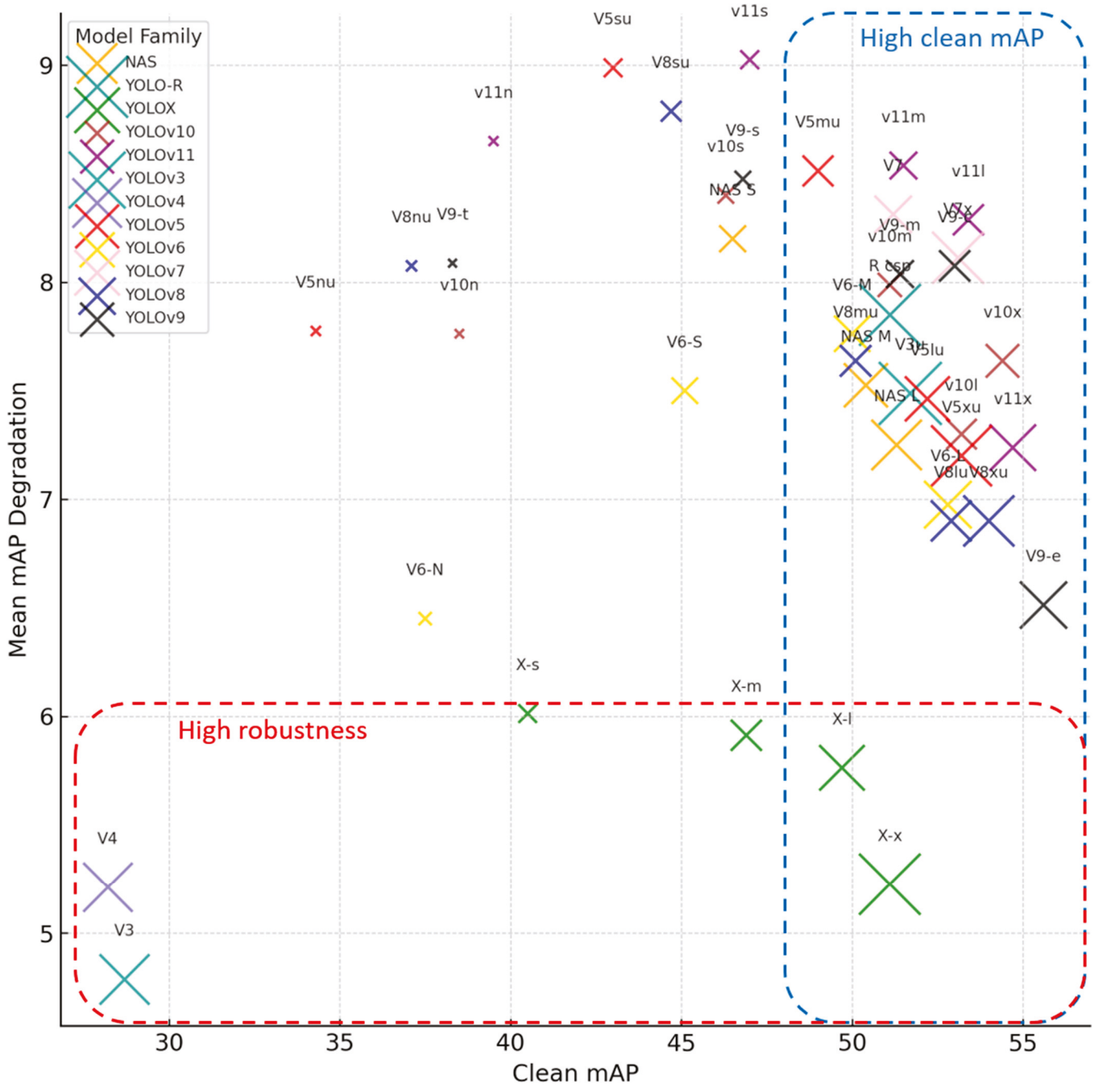


Figure 10. The average degradation across all attacks with clean mAP.

Decoupled head: YOLOX is the first YOLO variant to adopt a decoupled head architecture, separating classification and localization into independent branches. This design improves robustness by minimizing task interference—allowing each head to specialize without being impacted by conflicting gradients. Under adversarial conditions or image corruptions, this separation helps the model maintain more stable predictions, as it reduces

the risk of compounded errors due to misalignment between object classification and localization. Although later YOLO models also adopt decoupled heads, our results indicate that they do not achieve comparable robustness to YOLOX, suggesting that decoupling alone is not sufficient, but plays a complementary role within YOLOX's overall architecture.

Strong Data Augmentation: YOLOX employs extensive data augmentation techniques during training, such as Mosaic and MixUp, which expose the model to a broader distribution of visual patterns. This improves robustness by helping the model generalize across different lighting, textures, and occlusion conditions—characteristics that are common in real-world noise and perturbations. However, as observed in our experiments, subsequent models like YOLOv7 also utilize strong data augmentation but do not exhibit similar robustness under adversarial attacks. This implies that while augmentation enhances generalization, it does not fully explain YOLOX's superior resilience to perturbations and adversarial input.

Anchor-free Training: YOLOX replaces traditional anchor-based detection with an anchor-free mechanism, which simplifies the object detection process by eliminating predefined anchor box configurations. This not only reduces computational complexity but also improves robustness by avoiding overfitting to fixed anchor sizes and aspect ratios. Such flexibility enables the model to better detect objects that are distorted or misaligned due to adversarial noise or corruptions. Nonetheless, our experimental results show that models like YOLOv6 and YOLOv8, which also adopt anchor-free training, fail to match YOLOX's robustness. This suggests that anchor-free design, while helpful, is likely not the primary factor contributing to YOLOX's performance in adverse conditions.

Multi-Positives: YOLOX incorporates a multi-positive label assignment strategy, which allows multiple predicted boxes to be matched to a single ground truth object during training. This is particularly beneficial in cluttered scenes or when objects are partially occluded—scenarios where standard one-to-one matching may fail. By learning from multiple correct predictions, the model becomes more resilient to variations in object positioning and overlap. However, this technique is also used in later models like YOLOv6 and v7, which do not achieve the same robustness performance in our evaluations. Therefore, multi-positive assignment contributes to robustness but likely serves as a supporting mechanism rather than the core driver of YOLOX's resilience.

SimOTA: SimOTA, or Simple Optimal Transport Assignment, is the only feature of YOLOX that other YOLO models do not use. SimOTA works by assigning predicted bounding boxes to ground truth objects using a cost matrix that evaluates each prediction based on two factors: IoU (Intersection over Union), which measures spatial alignment, and classification confidence, which reflects the likelihood of a correct class match. It calculates the total cost for each possible match, dynamically selecting only the top-k predictions with the lowest cost for each ground truth object. This dynamic top-k thresholding lets SimOTA adapt to the specific quality of predictions in each image, focusing training on the highest-quality matches. This selective matching helps the model learn only from the clearest examples, improving accuracy and generalization. By combining classification confidence (how likely the prediction represents the correct object) with spatial alignment (measured through Intersection over Union, or IoU), SimOTA creates a balanced and adaptive matching process. Unlike traditional methods that use fixed matching criteria, SimOTA employs dynamic thresholding, allowing it to select only the highest-quality predictions for training, based on the specific characteristics of each image. This adaptability ensures that the model learns from the clearest, most reliable examples, reducing the influence of noisy or ambiguous predictions. As a result, the model learns representations that capture essential, generalizable features of objects, making it more effective when deployed in varied real-world scenarios. This dual focus on quality and spatial accuracy

makes SimOTA particularly robust, equipping models to handle new environments and unexpected variations in object appearance with improved accuracy and reliability.

In summary, although later YOLO models adopt several of YOLOX's innovations, only YOLOX consistently demonstrates superior robustness. This implies that robustness is not the result of any single feature but likely emerges from the specific combination of architectural and methodological choices—with SimOTA potentially being the most impactful based on our findings.

Also, it is observed that some small models behave better in some types of noise. In pixelate noise, motion blur, and defocus blur, the nano versions of v5, v6, v8, v10, and v11 present significant improvement in degradation than the bigger corresponding version. This may be because larger models with more parameters have a higher capacity to learn fine details in training data. While this allows them to capture intricate patterns in clean images, it also makes them more sensitive to high-frequency details and pixel-level structures. While larger model capacity generally leads to greater robustness against adversarial attacks, this is not necessarily the case for image corruption. This also can be shown in Figure 6, which illustrates the overall performance of each model in relation to its corresponding size.

Despite their overall robustness, even the top models experience variability in their resilience based on the type of corruption. Pixelation, in particular, leads to a notable drop in performance for many models, but YOLOX-x and YOLOX-l handle it relatively well compared to others. This indicates that while robustness is high in general, certain noise types still pose challenges, and specialized defenses might be needed for pixelation-type noises. Based on these results, YOLOX models, especially in their larger variants, are likely the best choice for applications requiring high robustness against adversarial attacks and corruption. While YOLOv3 and YOLOv4 could still be considered viable options, their relatively poor performance in clean images suggests they may not be the optimal choice for applications where accuracy is a critical requirement. Since YOLOv3 and YOLOv4 already perform relatively poorly on clean images compared to more advanced models, the degree of degradation they experience under various corruptions may appear less pronounced. In terms of architectural foundations, YOLOX incorporates components that were already explored in previous YOLO versions, but it also introduces distinct advancements that appear to enhance its robustness. YOLOX leverages CSPNet as its backbone—a structure that is also used in YOLOv5—while utilizing an FPN-based neck, as found in YOLOv3 and YOLOv3u. Despite these commonalities, YOLOv5 and YOLOv3u do not exhibit notable robustness against adversarial attacks and image corruption, suggesting that robustness in YOLOX is likely attributed to additional factors beyond these shared architectural elements. YOLOX builds upon YOLOv3 but makes five critical modifications: (1) decoupled head, (2) strong data augmentation, (3) anchor-free training, (4) multi-positives, and (5) SimOTA. To investigate the underlying factors contributing to the robustness of YOLOX, an analysis of the key architectural and methodological improvements it introduces is conducted hereafter.

Almost all models exhibit an upward trend in performance as their model size increases, with a characteristic curve after their smallest version (Figure 11). However, YOLOX deviates from this pattern, showing a downward trend in performance as model size increases. Additionally, smaller models generally achieve relatively low mean Average Precision (mAP) scores. Small models may inherently show limited mAP improvement with further reductions in model size because their baseline performance is already constrained by their reduced capacity. These models, designed with fewer parameters and simpler architectures, often lack the depth and feature extraction ability needed to capture intricate patterns in the data. Consequently, they start with relatively low mAP, meaning there is less room for a noticeable drop.

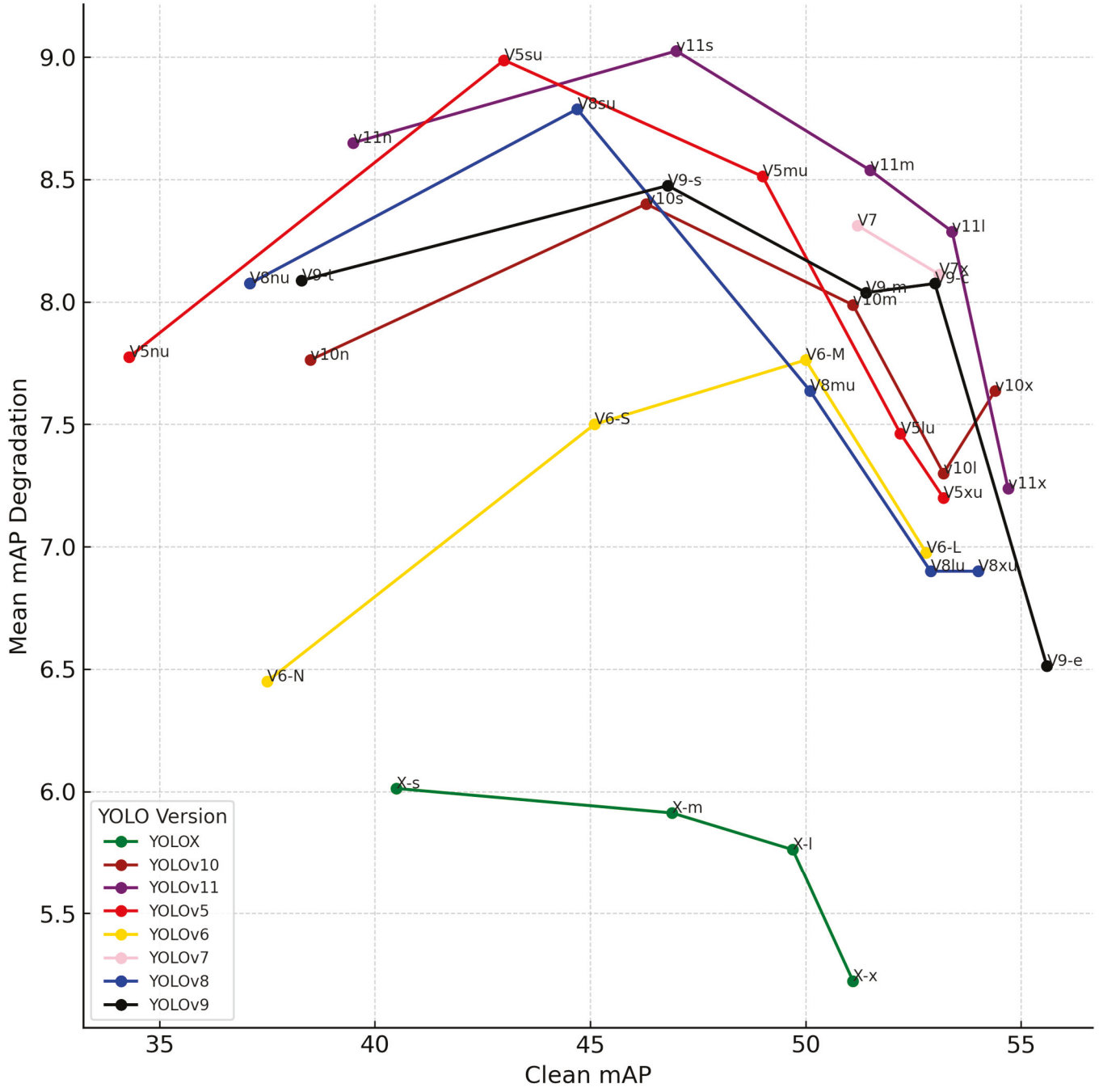


Figure 11. Model performance by size.

A limitation of our study is that we evaluated pre-trained models on a fixed test set without retraining. As a result, conventional statistical significance tests or confidence interval analyses could not be directly applied, since model-level variance across multiple training runs was not available. Future work involving retraining models from scratch or using cross-validation would allow for more rigorous statistical validation of the observed robustness differences.

Additionally, the scope of adversarial attacks in this study was limited to FGSM and PGD due to their wide adoption and interpretability, particularly in benchmarking scenarios. More recent and powerful attacks, such as AutoAttack, APGD, or Square Attack, were not included, mainly due to the high computational cost associated with testing a large number of YOLO variants. Moreover, our experiments focused on a black-box threat model,

which better reflects real-world conditions where model internals are typically inaccessible to attackers. However, we acknowledge that evaluating under white-box settings and incorporating stronger adversarial attacks would offer a more comprehensive assessment of model robustness. We consider this an important direction for future research.

7. Conclusions

This work has explored the robustness of YOLO models under adversarial attacks and image corruption, focusing on key architectural advancements and their impact on model resilience. YOLO models are widely used in fields such as autonomous driving, surveillance, and medical imaging, where even slight performance degradation under challenging conditions—such as adversarial attacks or image corruption—could lead to significant risks or errors. Ensuring that YOLO models are robust is not just a technical goal but a practical necessity for these high-stakes environments. Among the models reviewed, YOLOX emerges as the most robust, demonstrating a superior ability to retain performance in the face of adversarial perturbations and noisy data. This robustness may be largely attributed to the integration of the SimOTA algorithm, which optimizes label assignment and enhances model generalization by prioritizing high-quality predictions during training. By underscoring the robustness of YOLOX and identifying SimOTA as a potential key factor, this paper offers a foundation for future YOLO development focused on enhancing resilience against real-world adversarial conditions. Further investigation into SimOTA and similar algorithms could unlock new pathways to improve YOLO models, achieving a more effective balance among accuracy, speed, and robustness.

Author Contributions: Conceptualization, K.D.A. and G.A.P.; methodology, K.D.A. and G.A.P.; software, K.D.A.; validation, K.D.A. and G.A.P.; formal analysis, K.D.A.; investigation, K.D.A.; resources, K.D.A.; data curation, K.D.A.; writing—original draft preparation, K.D.A.; writing—review and editing, G.A.P.; visualization, K.D.A.; supervision, G.A.P.; project administration, G.A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in the dataset COCO at https://cocodataset.org/#home (accessed on 11 March 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
COCO	Common Object in Context
DNN	Deep neural network
CNN	Convolutional neural network
FPN	Feature Pyramid Network
PANet	Path Aggregation Networks
R-CNN	Region-based CNN
DETR	Detection Transformer
SSD	Single Shot Detection
FGSM	Fast Gradient Sign Method
PGD	Projected Gradient Descent
CSP	Cross-stage spatial connections
SAM	Spatial Attention Module
IoU	Intersection over Union
AP	Average Precision

NMS	Non-Maximum Suppression
BN	Batch normalization
EMA	Exponential Moving Average
SimOTA	Simplified Optimal Transport Assignment
QAT	Quantization
PTQ	Post-Training Quantization
ELAN	Efficient Layer Aggregation Network
GELAN	Generalized Efficient Layer Aggregation Network
PGI	Programmable Gradient Information
ART	Adversarial Robustness Toolbox
SPP	Spatial pyramid pooling
DFL	Distribution Focal Loss
IF	Information Bottleneck
CIB	Compact Inverted Block
PSA	Partial Self-Attention

References

1. Liang, S.; Wu, H.; Zhen, L.; Hua, Q.; Garg, S.; Kaddoum, G.; Hassan, M.M.; Yu, K. Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles. *IEEE Trans. Intell. Transport. Syst.* **2022**, *23*, 25345–25360. [CrossRef]
2. Yue, X.; Li, H.; Shimizu, M.; Kawamura, S.; Meng, L. YOLO-GD: A Deep Learning-Based Object Detection Algorithm for Empty-Dish Recycling Robots. *Machines* **2022**, *10*, 294. [CrossRef]
3. Zhu, J.; Li, X.; Jin, P.; Xu, Q.; Sun, Z.; Song, X. MME-YOLO: Multi-Sensor Multi-Level Enhanced YOLO for Robust Vehicle Detection in Traffic Surveillance. *Sensors* **2021**, *21*, 27. [CrossRef] [PubMed]
4. Qureshi, R.; Ragab, M.G.; Abdulkader, S.J.; Muneer, A.; Alqushaib, A.; Sumiea, E.H.; Alhussian, H. A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023). *IEEE Access* **2024**, *12*, 57815–57836.
5. Alif, M.A.R.; Hussain, M. YOLOv1 to YOLOv10: A Comprehensive Review of YOLO Variants and Their Application in the Agricultural Domain. *arXiv* **2024**, arXiv:2406.10139.
6. Mushtaq, M.; Akram, M.U.; Alghamdi, N.S.; Fatima, J.; Masood, R.F. Localization and Edge-Based Segmentation of Lumbar Spine Vertebrae to Identify the Deformities Using Deep Learning Models. *Sensors* **2022**, *22*, 1547. [CrossRef]
7. Baccouche, A.; Garcia-Zapirain, B.; Castillo Olea, C.; Elmaghraby, A.S. Breast Lesions Detection and Classification via YOLO-Based Fusion Models. *Comput. Mater. Contin.* **2021**, *69*, 1407–1425. [CrossRef]
8. Dyrmann, M.; Mortensen, A.K.; Linneberg, L.; Høye, T.T.; Bjerge, K. Camera Assisted Roadside Monitoring for Invasive Alien Plant Species Using Deep Learning. *Sensors* **2021**, *21*, 6126. [CrossRef]
9. Wang, Q.; Cheng, M.; Huang, S.; Cai, Z.; Zhang, J.; Yuan, H. A Deep Learning Approach Incorporating YOLO v5 and Attention Mechanisms for Field Real-Time Detection of the Invasive Weed *Solanum Rostratum* Dunal Seedlings. *Comput. Electron. Agric.* **2022**, *199*, 107194. [CrossRef]
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
11. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
12. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part I 14*; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. [CrossRef]
15. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
16. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**, arXiv:1905.11946.
17. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2020.

18. Apostolidis, K.D.; Papakostas, G.A. A Survey on Adversarial Deep Learning Robustness in Medical Image Analysis. *Electronics* **2021**, *10*, 2132. [CrossRef]
19. Michaelis, C.; Mitzkus, B.; Geirhos, R.; Rusak, E.; Bringmann, O.; Ecker, A.S.; Bethge, M.; Brendel, W. Benchmarking Robustness in Object Detection: Autonomous Driving When Winter Is Coming. *arXiv* **2019**, arXiv:1907.07484.
20. Maliamanis, T.; Papakostas, G. Adversarial Computer Vision: A Current Snapshot. In Proceedings of the Twelfth International Conference on Machine Vision (ICMV 2019); Osten, W., Nikolaev, D.P., Eds.; SPIE: Amsterdam, The Netherlands, 2020; p. 121.
21. Hussain, M. YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO. *IEEE Access* **2024**, *12*, 42816–42833. [CrossRef]
22. Terven, J.; Cordova-Esparza, D. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
23. Hussain, M.; Khanam, R. In-Depth Review of YOLOv1 to YOLOv10 Variants for Enhanced Photovoltaic Defect Detection. *Solar* **2024**, *4*, 351–386. [CrossRef]
24. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.
25. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2019**, arXiv:1706.06083.
26. Jain, S. Adversarial Attack on Yolov5 for Traffic and Road Sign Detection. In Proceedings of the 2024 4th International Conference on Applied Artificial Intelligence (ICAPAI), Halden, Norway, 16 April 2024; pp. 1–5.
27. Dai, L.; Wang, J.; Yang, B.; Chen, F.; Zhang, H. An Adversarial Example Attack Method Based on Predicted Bounding Box Adaptive Deformation in Optical Remote Sensing Images. *PeerJ Comput. Sci.* **2024**, *10*, e2053. [CrossRef]
28. Bayer, J.; Becker, S.; Münch, D.; Arens, M. Eigenpatches—Adversarial Patches from Principal Components. In Proceedings of the Advances in Visual Computing; Bebis, G., Ghiasi, G., Fang, Y., Sharf, A., Dong, Y., Weaver, C., Leo, Z., LaViola, J.J., Jr., Kohli, L., Eds.; Springer Nature: Cham, Switzerland, 2023; pp. 274–284.
29. Shack, J.; Petrovic, K.; Saukh, O. Breaking the Illusion: Real-World Challenges for Adversarial Patches in Object Detection. *arXiv* **2024**, arXiv:2410.19863.
30. Lian, J.; Pan, J.; Wang, L.; Wang, Y.; Chau, L.-P.; Mei, S. PADetBench: Towards Benchmarking Physical Attacks against Object Detection. *arXiv* **2024**, arXiv:2408.09181.
31. Liu, J.; Wang, Z.; Ma, L.; Fang, C.; Bai, T.; Zhang, X.; Liu, J.; Chen, Z. Benchmarking Object Detection Robustness against Real-World Corruptions. *Int. J. Comput. Vis.* **2024**, *132*, 4398–4416. [CrossRef]
32. Arsenos, A.; Karampinis, V.; Petrongonas, E.; Skliros, C.; Kollias, D.; Kollias, S.; Voulodimos, A. Common Corruptions for Enhancing and Evaluating Robustness in Air-to-Air Visual Object Detection. *arXiv* **2024**, arXiv:2405.06765. [CrossRef]
33. Ding, Y.; Luo, X. SDNIA-YOLO: A Robust Object Detection Model for Extreme Weather Conditions. *arXiv* **2024**, arXiv:2406.12395.
34. The PASCAL Visual Object Classes Challenge 2007 (VOC2007). Available online: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/ (accessed on 7 November 2024).
35. The PASCAL Visual Object Classes Challenge 2012 (VOC2012). Available online: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/ (accessed on 7 November 2024).
36. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part V*; Springer International Publishing: Cham, Switzerland, 2014.
37. Nicolae, M.-I.; Sinn, M.; Tran, M.N.; Buesser, B.; Rawat, A.; Wistuba, M.; Zantedeschi, V.; Baracaldo, N.; Chen, B.; Ludwig, H.; et al. Adversarial Robustness Toolbox v1.0.0. *arXiv* **2019**, arXiv:1807.01069.
38. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. *arXiv* **2014**, arXiv:1409.4842.
39. [1312.4400] Network in Network. Available online: https://arxiv.org/abs/1312.4400 (accessed on 7 November 2024).
40. [1612.08242] YOLO9000: Better, Faster, Stronger. Available online: https://arxiv.org/abs/1612.08242 (accessed on 7 November 2024).
41. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
42. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
44. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
45. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.

46. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
47. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995.
48. Wang, C.-Y.; Mark Liao, H.-Y.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
50. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11211, pp. 3–19. ISBN 978-3-030-01233-5.
51. Yun, S.; Han, D.; Chun, S.; Oh, S.J.; Yoo, Y.; Choe, J. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6022–6031.
52. Ghiasi, G.; Lin, T.-Y.; Le, Q.V. DropBlock: A Regularization Method for Convolutional Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Montréal, QC, Canada, 2018; Volume 31.
53. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *AAAI* **2020**, *34*, 12993–13000. [CrossRef]
54. Ultralytics YOLOv5. Available online: https://docs.ultralytics.com/models/yolov5 (accessed on 7 November 2024).
55. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. You Only Learn One Representation: Unified Network for Multiple Tasks. *arXiv* **2021**, arXiv:2105.04206.
56. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
57. Ge, Z.; Liu, S.; Li, Z.; Yoshie, O.; Sun, J. OTA: Optimal Transport Assignment for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
58. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
59. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-Style ConvNets Great Again. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13728–13737.
60. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-Aligned One-Stage Object Detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3490–3499.
61. Zhang, H.; Wang, Y.; Dayoub, F.; Sunderhauf, N. VarifocalNet: An IoU-Aware Dense Object Detector. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8510–8519.
62. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
63. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
64. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
65. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H. Designing Network Design Strategies Through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800.
66. Ultralytics YOLOv8. Available online: https://docs.ultralytics.com/models/yolov8 (accessed on 7 November 2024).
67. Ultralytics YOLO-NAS (Neural Architecture Search). Available online: https://docs.ultralytics.com/models/yolo-nas (accessed on 7 November 2024).
68. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *European Conference on Computer Vision*; Springer Nature Switzerland: Cham, Switzerland, 2024; pp. 1–21.
69. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.

70. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling Up Your Kernels to 31x31: Revisiting Large Kernel Design in CNNs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
71. YOLO11 🚀 NEW—Ultralytics YOLO Docs. Available online: https://docs.ultralytics.com/models/yolo11/ (accessed on 7 November 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Review

Cognitive Bias Mitigation in Executive Decision-Making: A Data-Driven Approach Integrating Big Data Analytics, AI, and Explainable Systems

Leonidas Theodorakopoulos, Alexandra Theodoropoulou and Constantinos Halkiopoulos *

Department of Management Science and Technology, University of Patras, 26334 Patras, Greece; theodleo@upatras.gr (L.T.); theodoropouloua@upatras.gr (A.T.)
* Correspondence: halkion@upatras.gr

Abstract: Cognitive biases continue to pose significant challenges in executive decision-making, often leading to strategic inefficiencies, misallocation of resources, and flawed risk assessments. While traditional decision-making relies on intuition and experience, these methods are increasingly proving inadequate in addressing the complexity of modern business environments. Despite the growing integration of big data analytics into executive workflows, existing research lacks a comprehensive examination of how AI-driven methodologies can systematically mitigate biases while maintaining transparency and trust. This paper addresses these gaps by analyzing how big data analytics, artificial intelligence (AI), machine learning (ML), and explainable AI (XAI) contribute to reducing heuristic-driven errors in executive reasoning. Specifically, it explores the role of predictive modeling, real-time analytics, and decision intelligence systems in enhancing objectivity and decision accuracy. Furthermore, this study identifies key organizational and technical barriers—such as biases embedded in training data, model opacity, and resistance to AI adoption—that hinder the effectiveness of data-driven decision-making. By reviewing empirical findings from A/B testing, simulation experiments, and behavioral assessments, this research examines the applicability of AI-powered decision support systems in strategic management. The contributions of this paper include a detailed analysis of bias mitigation mechanisms, an evaluation of current limitations in AI-driven decision intelligence, and practical recommendations for fostering a more data-driven decision culture. By addressing these research gaps, this study advances the discourse on responsible AI adoption and provides actionable insights for organizations seeking to enhance executive decision-making through big data analytics.

Keywords: decision intelligence; algorithmic bias; artificial intelligence; strategic reasoning; real-time analytics; AI-powered decision support; data-driven strategy; explainable AI (XAI)

1. Introduction

Cognitive biases represent a significant challenge in executive decision-making within today's complex and high-stakes business environment. Corporate leaders face a multifaceted decision landscape influenced by organizational culture, historical precedent, external pressures, and deeply embedded psychological tendencies that often operate below conscious awareness [1]. Despite the traditional reverence for intuition and experience in executive circles, research has increasingly demonstrated how these cognitive shortcuts

can lead to systematic errors in judgment, resulting in suboptimal strategic outcomes, resource misallocation, and missed market opportunities [2].

The impact of these cognitive distortions is particularly pronounced in executive contexts, where confirmation bias reinforces existing assumptions, anchoring bias tethers decisions to initial reference points, and overconfidence bias inflates the perceived accuracy of strategic forecasts. These psychological tendencies collectively undermine rational decision-making processes and can propagate throughout organizations, affecting everything from capital allocation to talent management and competitive strategy [2].

The emergence of big data analytics and artificial intelligence (AI) represents a paradigm shift in how organizations approach strategic decision-making. For the first time, executives have access to tools that can systematically counteract cognitive limitations through data-driven methodologies. Machine learning algorithms, predictive analytics, and real-time data processing capabilities enable businesses to identify patterns beyond human perception, challenge entrenched assumptions, and augment decision-making with objective insights derived from comprehensive datasets [3]. Advanced analytical approaches such as explainable AI (XAI), causal inference modeling, and anomaly detection further enhance the potential for more rational, evidence-based strategic decisions.

However, integrating data-driven insights into executive workflows presents significant challenges. Questions regarding data reliability, model interpretability, organizational readiness, and ethical implications must be addressed for analytics to mitigate cognitive biases effectively. The path from traditional intuition-based leadership to data-augmented decision-making requires careful navigation of technical and cultural barriers.

1.1. Literature Review Methodology and Current Research Landscape

This review employed a systematic approach to identify relevant literature at the intersection of cognitive bias mitigation and big data analytics in decision-making contexts. We searched the Web of Science, Scopus, and IEEE Xplore databases using the keywords (“cognitive bias*” OR “decision bias*”) AND (“big data” OR “artificial intelligence” OR “machine learning”) AND (“decision making” OR “executive*” OR “management”) for publications from 2018–2024. After screening 847 initial results, 156 studies met inclusion criteria focusing on empirical applications of AI/analytics for bias reduction in organizational settings.

Table 1 synthesizes ten (10) pivotal studies that exemplify current understanding of how analytical technologies influence executive decision processes, selected based on citation impact, methodological rigor, and relevance to bias mitigation applications.

Research Gaps Identified: Current literature reveals three critical gaps: (1) limited empirical validation of bias reduction effectiveness in real-world executive contexts, (2) insufficient integration of explainable AI techniques for maintaining decision transparency, and (3) lack of comprehensive frameworks addressing both technical and organizational implementation challenges.

Building on this foundation, our research examines multiple dimensions of the bias mitigation challenge. Figure 1 presents the conceptual framework that guides our investigation, illustrating the interconnected nature of cognitive biases, analytical methodologies, and organizational factors that collectively shape executive decision quality.

Table 1. Summary of key studies on big data analytics and cognitive bias mitigation.

References	Focus Area	Key Findings
Rastogi et al. (2022) [4]	Data-driven decision-making biases	Large-scale datasets can enhance accuracy but may reinforce biases through flawed interpretations
Baudel et al. (2020) [5]	Automated bias detection	ABI Approach using Cumulative Prospect Theory effectively detects risk-seeking biases in business contexts
Haag et al. (2024) [6]	XAI for anchoring bias	Explainable AI systems can mitigate anchoring bias while maintaining user trust through transparency
Rastogi et al. (2022) [7]	Human-AI collaboration	Time-based strategies effectively mitigate confirmation and availability biases in collaborative settings
Power et al. (2019) [8]	Decision support systems	AI-powered DSSs can both enhance and distort decision-making depending on implementation approach
Hamdam et al. (2022) [9]	Government sector applications	Big Data Analytics Capability significantly improves decision-making through empirical validation
Acciarini et al. (2021) [10]	Intelligence analysis	Serious games (RECOBIA & LEILA) effectively train analysts to recognize and overcome cognitive biases
Wang et al. (2019) [11]	Evidence-based decision-making	Critical thinking and structured processes essential for effective bias mitigation through analytics
Polonioli et al. (2023) [12]	Audit decision-making	Data visualization integration can create biases due to information overload if not properly managed
Deng et al. (2023) [13]	Strategic decision-making	Four-phase model (Analysis, Decision, Onboarding, Control) effectively mitigates biases in uncertain environments

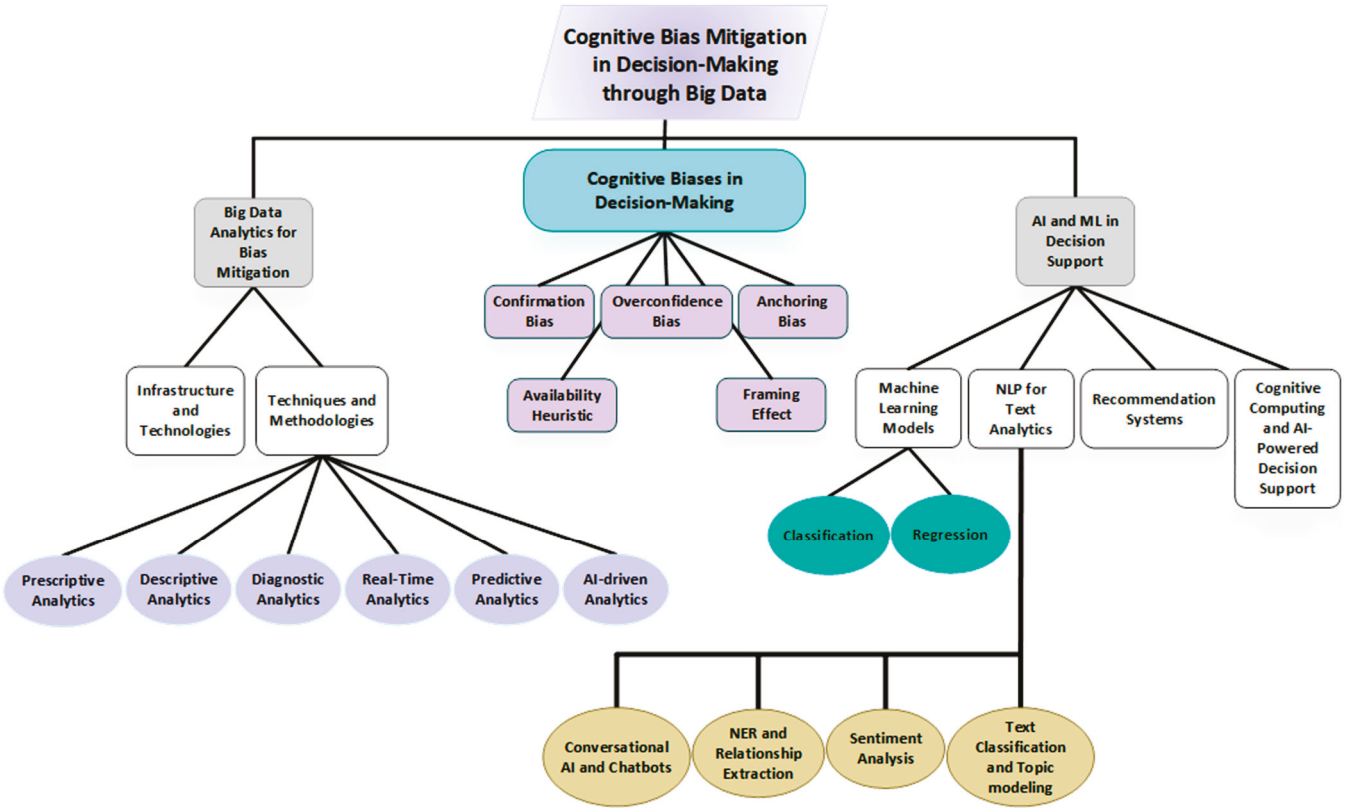


Figure 1. Overview of surveyed key topics.

1.2. Research Objectives and Contributions

This study addresses identified gaps by examining mechanisms through which big data analytics, AI, and XAI can systematically mitigate cognitive biases in executive decision-making while maintaining accountability and organizational acceptance.

Primary Research Questions:

1. How do specific AI/ML techniques detect and counteract different types of cognitive biases in executive decision contexts?
2. What are the key technical and organizational barriers to implementing bias mitigation systems?
3. How can explainable AI enhance trust and adoption of automated debiasing tools?

Novel Contributions:

- Comprehensive taxonomy of bias mitigation mechanisms across descriptive, predictive, and prescriptive analytics
- Empirical evaluation framework comparing A/B testing, simulation experiments, and behavioral assessments for measuring bias reduction effectiveness
- Implementation roadmap addressing both technical requirements and organizational change management for bias mitigation systems
- XAI integration model balancing algorithmic transparency with decision-making efficiency

Scope and Applications: While primarily focused on corporate executive decision-making, findings extend to healthcare administration, public policy formulation, and emergency management contexts where bias mitigation is critical for optimal outcomes.

1.3. Paper Structure

Section 2 establishes theoretical foundations, examining cognitive biases in executive contexts and big data analytics capabilities. Section 3 analyzes real-world manifestations of bias in strategic decision-making. Section 4 investigates bias mitigation mechanisms through analytics, including AI-driven detection and correction systems. Section 5 addresses implementation challenges and limitations. Section 6 explores emerging trends, XAI applications, and research gaps. Section 7 presents conclusions and practical recommendations for organizations seeking to enhance decision quality through data-driven bias mitigation.

2. Theoretical Foundations

2.1. Cognitive Biases in Executive Decision-Making

Cognitive biases represent systematic deviations from rational judgment that lead individuals to make suboptimal decisions based on mental shortcuts, preconceptions, or flawed heuristics [14]. Within executive decision-making contexts, these biases pose significant challenges as leaders operate under conditions of uncertainty, time pressure, and high stakes [15]. This section examines five key cognitive biases that substantially impact executive decision-making: confirmation bias, overconfidence bias, anchoring bias, availability heuristic, and framing effect [16] (Table 2).

Also, the below conceptual framework (Figure 2) illustrates the five primary cognitive biases that systematically distort executive decision-making processes.

At the center, "Cognitive Biases" represents the overarching psychological phenomena that influence how leaders process information and make strategic choices. Each surrounding node depicts a specific bias with its key characteristics:

Table 2. Overview of Key Cognitive Biases in Executive Decision-Making.

Cognitive Bias	Definition	Executive Impact	Mitigation Approaches	Key References
Confirmation Bias	Tendency to seek and interpret information that confirms preexisting beliefs while dismissing contradictory evidence	Strategic blind spots, escalation of commitment to failing projects, resistance to market changes	Devil's advocacy procedures, structured decision processes, AI-driven analytics	[17–48]
Availability Heuristic	Overvaluing easily recalled or recent information	Distorted risk assessment, reactive strategies based on memorable events	Statistical analysis, systematic data review, AI-driven trend analysis	[49–57]
Framing Effect	Decision influence by how information is presented (gains vs. losses)	Inconsistent risk preferences, communication effectiveness variations	Multi-perspective analysis, standardized metrics, numerical frameworks	[58–72]
Overconfidence Bias	Excessive confidence in one's knowledge, predictive ability, or control over outcomes	Unrealistic project timelines, excessive M&A activity, underestimation of risks	Probabilistic reasoning, Monte Carlo simulations, analytical validation	[30–34,73–80]
Anchoring Bias	Over-reliance on initial information when making subsequent judgments	Distorted financial planning, suboptimal negotiations, biased performance evaluations	Multiple independent estimates, first-principles reasoning, algorithmic baselines	[35–48,81–93]

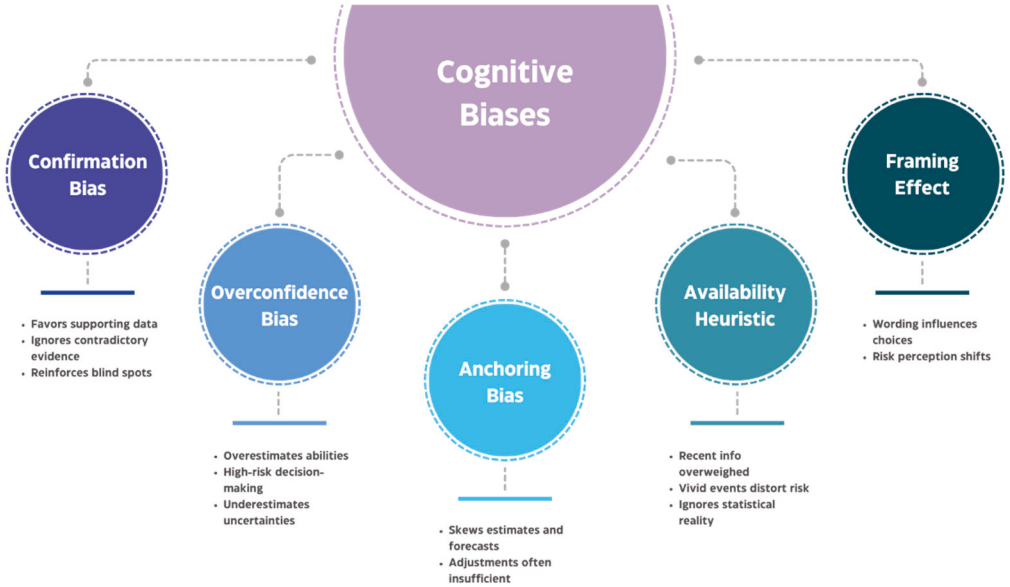


Figure 2. The five cognitive biases in executive decision-making.

Confirmation Bias (left) manifests when executives favor information supporting their existing beliefs while ignoring contradictory evidence, leading to strategic blind spots and resistance to market changes.

Overconfidence Bias (lower left) reflects executives' tendency to overestimate their abilities and underestimate risks, resulting in unrealistic project timelines and excessive risk-taking in areas such as mergers and acquisitions.

Anchoring Bias (bottom) occurs when initial information disproportionately influences subsequent decisions, affecting financial forecasts and negotiations even when the anchor proves arbitrary or outdated.

Availability Heuristic (lower right) leads executives to overweight easily recalled events, causing distorted risk assessments based on memorable incidents rather than statistical reality.

Framing Effect (right) demonstrates how identical information presented differently can reverse risk preferences, influencing strategic communications and decision outcomes.

The dotted connections between biases indicate their interconnected nature—these cognitive distortions often reinforce each other, creating compound effects that further compromise decision quality. Understanding these relationships is essential for developing comprehensive mitigation strategies through big data analytics, as discussed in subsequent sections.

2.1.1. Confirmation Bias

Confirmation bias represents the tendency to search for, interpret, and recall information that confirms preexisting beliefs while giving disproportionately less consideration to alternative possibilities [17]. In executive contexts, this bias manifests through selective information processing, where leaders unconsciously filter data to support their strategic vision while dismissing contradictory evidence [18]. The hierarchical structure of organizations often amplifies this bias through "filtering bias," where subordinates selectively present information they believe executives want to hear [19].

This cognitive distortion particularly affects strategic planning and risk assessment processes. Executive teams with entrenched assumptions about market stability often dismiss early disruption signals as temporary anomalies rather than fundamental shifts in competitive dynamics [20]. The escalation of commitment to failing projects exemplifies confirmation bias in action, as executives seek evidence supporting initial decisions rather than objectively reassessing situations [21]. Historical analyses have documented how established retailers failed to recognize e-commerce threats despite clear market data indicating shifting consumer preferences, demonstrating the profound impact of this bias on strategic adaptation [22–24].

Effective mitigation strategies encompass institutionalized devil's advocacy procedures that systematically challenge existing assumptions [25–27], structured decision-making processes requiring explicit documentation of reasoning and evidence [28], and AI-driven analytics that provide objective, data-based insights independent of human preconceptions [29].

2.1.2. Overconfidence Bias

Overconfidence bias involves excessive confidence in one's abilities, knowledge, or chances of success, leading to systematic underestimation of risks and overestimation of positive outcomes [73]. This bias appears particularly pronounced among executives, where past successes and positional authority reinforce beliefs in personal judgment capabilities [74].

In corporate settings, overconfidence manifests through multiple channels. Executives frequently establish unrealistic project timelines, reflecting the "planning fallacy" wherein best-case scenarios dominate planning processes while potential obstacles receive insufficient consideration [75]. The merger and acquisition domain provides particularly compelling evidence, with overconfident executives pursuing excessive acquisition activity based on inflated synergy estimates and underestimation of integration complexities [76]. Strategic initiatives often suffer from inadequate risk assessment as leaders overestimate their ability to control market dynamics and competitive responses [77]. The psychological origins of this bias stem from self-serving attribution patterns, where executives attribute success to personal ability while externalizing blame for failures [78].

Evidence-based mitigation approaches incorporate systematic use of predictive analytics and Monte Carlo simulations to ground decisions in probabilistic reasoning rather than intuitive certainty [79]. Organizations that implement analytical validation paradigms within strategic planning cycles demonstrate improved calibration between confidence levels and actual outcomes [80]. Furthermore, cultivating organizational cultures that promote constructive skepticism and encourage challenging of executive assumptions has proven effective in moderating overconfidence effects [80].

2.1.3. Anchoring Bias

Anchoring bias occurs when individuals rely disproportionately on initial information when making subsequent judgments, even when that information proves arbitrary or irrelevant to the decision context [81]. In executive decision-making, anchoring significantly impacts financial planning, negotiations, and strategic goal-setting processes [82].

Revenue forecasting exemplifies this phenomenon, where executive teams often remain tied to initial projections despite evolving market conditions that warrant substantial revision [83]. Negotiation contexts reveal similar patterns, with opening offers establishing psychological reference points that disproportionately influence final agreements, regardless of objective valuation metrics [84]. Human resource decisions demonstrate anchoring effects when compensation determinations rely on historical salary levels rather than current market rates or performance metrics [85]. Strategic planning processes frequently exhibit anchoring when organizations set objectives based on competitors' historical performance rather than conducting independent potential analyses [86,87].

Effective countermeasures require systematic approaches to decision-making. Organizations benefit from obtaining multiple independent estimates before critical decisions, ensuring diverse perspectives challenge initial anchors [88]. Employing first-principles reasoning in budgeting processes helps teams evaluate expenditures based on fundamental requirements rather than incremental adjustments to potentially flawed baselines [89]. AI-based pricing algorithms that analyze current market dynamics rather than historical precedents offer technological solutions to bypass anchoring effects [90,91]. Additionally, conducting systematic post-mortem analyses enables organizations to identify where anchoring influenced past decisions and develop institutional awareness of this bias [92,93].

2.1.4. Availability Heuristic

The availability heuristic leads individuals to overestimate the probability of events that are easily recalled, typically because they are recent, emotionally salient, or widely publicized [49,50]. This cognitive bias causes executives to overweight memorable events in decision-making while undervaluing comprehensive statistical information that may provide more accurate risk assessments [51,52].

Business contexts reveal multiple manifestations of this bias. Risk assessment processes often become distorted when executives base probability estimates on vivid events such as highly publicized cyber-attacks or market crashes, rather than systematic analysis of actual occurrence frequencies [49,50]. Strategic planning suffers when recent successes or failures disproportionately influence future strategies, causing organizations to overreact to temporary fluctuations while missing long-term trends [51]. Resource allocation decisions driven by anecdotal evidence rather than comprehensive data analysis lead to suboptimal investment patterns [52]. These patterns create organizational feedback loops where memorable narratives override data-driven insights, perpetuating decision-making based on salience rather than statistical relevance [51,52].

Mitigation strategies leverage both technological and organizational approaches. AI-driven analytics shift focus from anecdotal evidence to statistical patterns by processing

comprehensive datasets that human cognition cannot effectively synthesize [53–55]. Structured decision-making techniques such as pre-mortem analyses force consideration of multiple scenarios beyond those readily available to memory [56]. Organizations that foster data-driven cultures prioritizing empirical evidence over memorable instances demonstrate improved calibration in risk assessment and strategic planning [57].

2.1.5. Framing Effect

The framing effect demonstrates how identical information presented in different formats can lead to opposing decisions, violating fundamental principles of rational choice [58]. According to prospect theory, individuals exhibit risk-averse behavior when outcomes are framed as gains but become risk-seeking when identical outcomes are framed as losses [59].

Executive communications and strategic choices reveal substantial framing effects. Budget proposals characterized as "efficiency enhancements" consistently receive greater organizational support than identical measures framed as "cost reductions," despite equivalent fiscal impacts [60]. Strategic initiatives described in terms of growth opportunities generate fundamentally different responses than those emphasizing threat mitigation, even when the underlying economic considerations remain constant [61,62].

Addressing framing effects requires deliberate cognitive strategies. Decision-makers must consciously evaluate choices from multiple perspectives, actively considering how different presentations might influence judgment [63,64]. Using precise numerical terminology rather than emotionally loaded language helps minimize linguistic framing effects [65]. Organizations benefit from implementing standardized decision frameworks that present information in consistent formats, reducing the influence of presentation variations on strategic choices [66].

2.2. Big Data Analytics: Framework and Applications

Big data analytics represents a comprehensive methodological framework for collecting, processing, and analyzing vast, complex datasets to extract actionable insights that enhance organizational decision-making [67]. This field has undergone rapid evolution alongside digital transformation, enabling organizations to leverage diverse data sources for strategic advantage [68,69].

2.2.1. Fundamental Characteristics: The 5 V's Framework

The conceptual foundation of big data analytics rests upon five key dimensions that distinguish it from traditional data processing approaches. Volume encompasses the unprecedented scale of modern data generation, with organizations routinely managing petabyte and exabyte-level datasets that exceed conventional storage and processing capabilities [67]. Variety reflects the heterogeneous nature of contemporary data, spanning structured formats conforming to predefined schemas, semi-structured data such as XML and JSON, and unstructured information including text, images, and multimedia content [67]. Velocity captures the accelerating pace of data generation and the corresponding requirement for real-time or near-real-time processing capabilities [67]. Veracity addresses critical concerns regarding data quality, accuracy, and trustworthiness, recognizing that analytical insights depend fundamentally on underlying data integrity [68]. Value represents the ultimate objective of big data initiatives: transforming raw information into meaningful insights that drive business success and competitive advantage [68].

2.2.2. Infrastructure and Technologies

Modern big data infrastructure comprises sophisticated technological ecosystems designed to manage the scale and complexity of contemporary data environments [69,70]. Storage systems have evolved from traditional relational databases to distributed architec-

tures capable of handling massive, heterogeneous datasets. The Hadoop Distributed File System (HDFS) provides fault-tolerant storage by distributing data across multiple nodes, ensuring high availability through redundancy mechanisms [71,72]. NoSQL databases offer specialized solutions for different data management requirements: document stores accommodate schema-less data, column-family databases optimize read-intensive operations, key-value stores provide exceptional retrieval speed, and graph databases excel at mapping complex relationships between entities [73].

Processing frameworks have similarly evolved to meet big data demands. While MapReduce initially revolutionized parallel processing, Apache Spark (all versions through 4.0.1) has emerged as the dominant framework, offering in-memory processing capabilities that dramatically improve performance for iterative algorithms and real-time analytics [71,72]. Stream processing platforms including Apache Kafka and Flink enable organizations to analyze data in motion, supporting use cases requiring immediate insights [69].

Cloud computing has fundamentally transformed big data infrastructure accessibility. Platforms from Amazon Web Services, Microsoft Azure, and Google Cloud provide scalable, cost-effective solutions that eliminate barriers to entry for sophisticated analytics [74,75]. Edge computing complements cloud architectures by processing data near its source, reducing latency for time-sensitive applications while improving data privacy [74,75]. Orchestration tools such as Apache Airflow and Kubernetes automate workflow management, while comprehensive security frameworks ensure compliance with regulations including GDPR and industry-specific requirements [76,77].

The architectural diagram (Figure 3) presents the multi-layered infrastructure supporting big data analytics for executive decision-making and bias mitigation. The visualization employs a hierarchical structure that traces data flow from initial collection through final analytical outputs, demonstrating the technological complexity underlying modern bias mitigation systems.

The Data Sources layer (top) identifies primary inputs including IoT devices, social media streams, machine logs, and transactional records, representing the diverse data ecosystem that feeds analytical processes. The Storage Layer illustrates the dual architecture of distributed file systems (HDFS) for large-scale data storage and specialized NoSQL databases optimized for different data structures—document-based systems for unstructured content, column-family databases for analytical workloads, key-value stores for rapid retrieval, and graph databases for relationship mapping [71–73].

The Processing Layer showcases the evolution from batch-oriented MapReduce to real-time processing frameworks. Apache Spark's (ver. 3.5.0) in-memory capabilities enable iterative machine learning algorithms essential for bias detection, while stream processing platforms (Kafka, Flink) support real-time bias identification in ongoing decisions [71,72]. The Orchestration Layer demonstrates how workflow management tools coordinate complex analytical pipelines, ensuring systematic bias assessment across multiple data streams.

The Analytics AI Layer represents where bias mitigation actually occurs, through machine learning algorithms that identify patterns beyond human perception, business intelligence tools that visualize bias indicators, and advanced analytics that prescribe corrective actions. The diagram shows these analytical capabilities deployed through Cloud Computing and Edge Computing/AI models, with the former providing scalable processing power and the latter enabling real-time bias detection at decision points [74,75].

Critical to the framework are Security Governance mechanisms ensuring data integrity and regulatory compliance, as biased or compromised data would undermine mitigation efforts [76,77]. The Future Technologies component acknowledges emerging

capabilities—quantum computing for complex optimization problems and AI-driven data management for automated bias detection in data preparation stages.

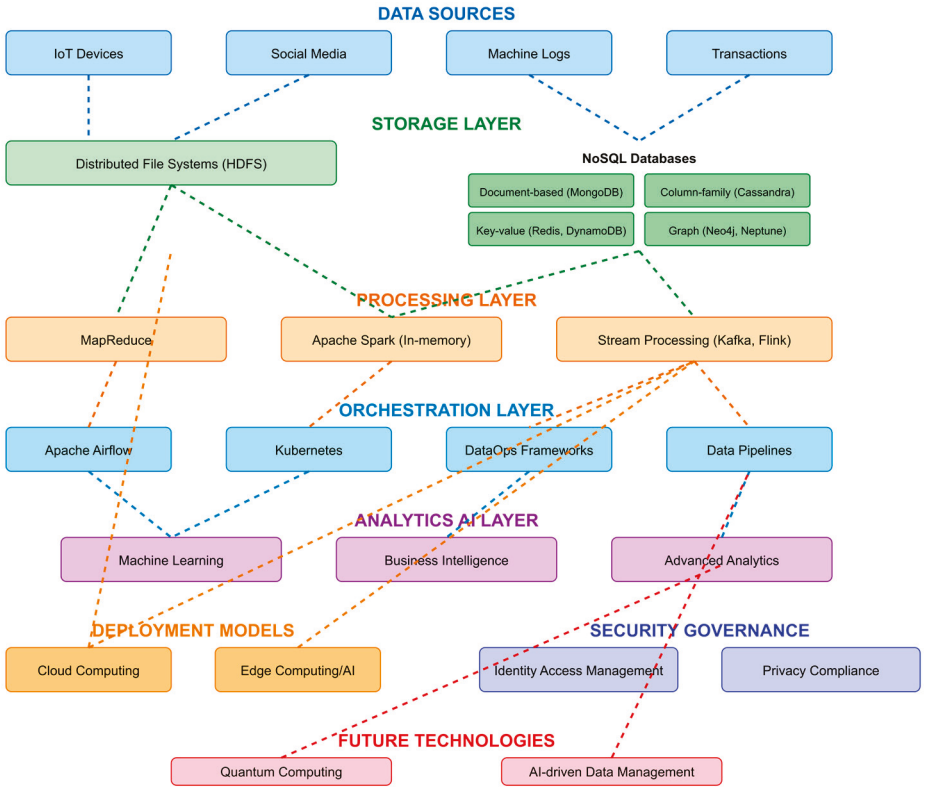


Figure 3. Big Data Infrastructure and Technologies Ecosystem.

The interconnecting arrows demonstrate data flow patterns, emphasizing that bias mitigation requires seamless integration across all layers. This comprehensive infrastructure enables the analytical methodologies discussed in Section 2.2.3 and supports the AI/ML applications detailed in Section 2.2.4, forming the technological foundation for systematic cognitive bias mitigation in executive decision-making.

2.2.3. Analytical Methodologies

Organizations employ six primary analytical approaches to extract value from big data, each serving distinct purposes within the analytical continuum [78]. Descriptive analytics provides historical perspective by summarizing past events and identifying patterns within existing data [79]. Diagnostic analytics extends this foundation by investigating causal relationships and uncovering factors driving observed outcomes [80]. Predictive analytics leverages statistical modeling and machine learning to forecast future trends based on historical patterns [81]. Prescriptive analytics advances beyond prediction to recommend optimal actions given specific constraints and objectives [82]. Real-time analytics enables immediate response to changing conditions through continuous data processing [83]. AI-driven analytics automates the discovery of complex patterns and relationships that may elude human analysis [84].

2.2.4. Artificial Intelligence and Machine Learning Integration

Artificial intelligence and machine learning technologies constitute the analytical engine of modern big data systems, enabling automated extraction of sophisticated insights from massive datasets [85,86]. Classification algorithms categorize data into predefined groups, supporting applications ranging from fraud detection to medical diagnosis [87,88].

Regression models predict continuous outcomes, facilitating sales forecasting, risk assessment, and demand planning [89]. Ensemble methods and deep learning architectures uncover complex, non-linear relationships within high-dimensional data [90].

Natural language processing has emerged as a critical capability for analyzing the vast quantities of unstructured text data generated by modern organizations [94]. Sentiment analysis extracts emotional content from customer feedback, social media, and other textual sources [95]. Named entity recognition identifies and extracts key information elements from documents [96]. Conversational AI enables natural language interactions between humans and analytical systems [97–99].

Advanced applications leverage these foundational technologies to create sophisticated analytical capabilities. Recommendation systems employ collaborative filtering and deep learning to personalize user experiences [100–103]. Cognitive computing platforms such as IBM Watson and Microsoft Azure AI simulate human reasoning processes, enabling complex decision support across domains including healthcare, finance, and legal services [104–111]. These systems demonstrate adaptive learning capabilities, continuously improving their performance through interaction with data and users [98,99].

2.2.5. Implications for Bias Mitigation

The integration of big data analytics with executive decision-making offers multiple mechanisms for cognitive bias mitigation. AI systems process data without susceptibility to emotional influences or hierarchical pressures that affect human judgment. Machine learning algorithms identify patterns within datasets too large and complex for human cognitive processing, revealing insights that challenge existing assumptions. Real-time analytics provides immediate feedback on decision outcomes, enabling rapid identification and correction of biased judgment patterns. Statistical models quantify uncertainty explicitly, countering human tendencies toward overconfidence and providing calibrated probability assessments.

However, these technologies also present significant challenges. Algorithmic bias may emerge when models trained on historical data perpetuate past discriminatory patterns. Model interpretability remains a critical concern, as complex algorithms may produce accurate predictions without transparent reasoning. Organizational resistance to data-driven approaches can limit the practical impact of analytical insights. These considerations underscore the importance of thoughtful implementation strategies that leverage technological capabilities while addressing inherent limitations.

2.3. Theoretical Integration

This theoretical foundation establishes a comprehensive framework for understanding the intersection between cognitive biases and big data analytics in executive decision-making. The systematic biases that affect human judgment create predictable distortions in strategic choices, while big data analytics offers powerful tools for introducing objectivity and empirical rigor into decision processes. The convergence of these domains presents both unprecedented opportunities and complex challenges for organizations seeking to enhance decision quality.

The effectiveness of analytics-based bias mitigation depends critically on recognizing that technology alone cannot eliminate cognitive distortions. Rather, successful implementation requires integrating analytical capabilities with organizational processes, cultural change, and human expertise. The subsequent sections of this manuscript examine specific mechanisms through which big data analytics can address cognitive biases, evaluate empirical evidence for effectiveness, and identify key implementation challenges that organizations must navigate to realize the full potential of data-driven decision enhancement.

The heat map below (Figure 4) illustrates the differential effectiveness of six analytical methodologies in mitigating five key cognitive biases identified in executive decision-making. The matrix synthesizes empirical evidence from the literature to provide a visual guide for selecting appropriate analytical interventions based on the specific bias being addressed.

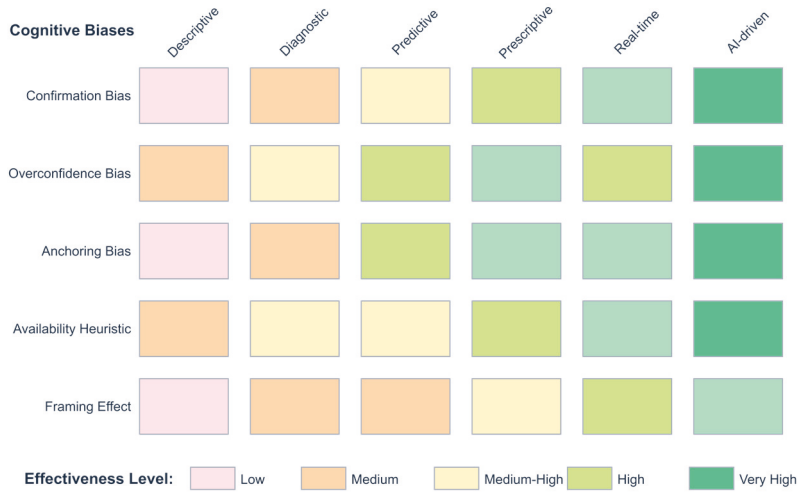


Figure 4. Bias-analytics effectiveness matrix.

The color gradient represents effectiveness levels ranging from low (light red) to very high (dark green). AI-driven analytics demonstrates the highest overall effectiveness across all biases, particularly in addressing confirmation bias and overconfidence, where its ability to process vast datasets without preconceptions provides maximum value. Prescriptive and real-time analytics show strong performance in mitigating anchoring and availability biases by providing immediate, optimization-based recommendations that bypass historical reference points.

Notably, descriptive analytics shows limited effectiveness across most biases, as it primarily reports historical patterns without challenging underlying assumptions. This finding underscores that merely presenting data is insufficient for bias mitigation; rather, advanced analytical techniques that actively identify patterns, generate predictions, and recommend actions are necessary to counteract deeply ingrained cognitive distortions.

The matrix reveals that no single analytical approach provides universal bias mitigation, suggesting that organizations should implement comprehensive analytical ecosystems combining multiple methodologies. The effectiveness ratings are based on empirical studies referenced in Section 2.1 (references [17–66] for bias characteristics) and Section 2.2 (references [78–84] for analytics capabilities), providing evidence-based guidance for practitioners designing bias mitigation strategies.

Finally, the below conceptual mind map (Figure 5) visualizes the interconnected domains that constitute the theoretical framework for bias mitigation through big data analytics. The central node represents the convergence of cognitive psychology and data science, with four primary branches illustrating the essential components of this integration.

The Cognitive Biases branch (red, upper left) identifies the five key biases examined in this study, with associated reference clusters indicating the depth of literature supporting each bias's characterization. The Analytics Technologies branch (blue, upper right) maps the technological infrastructure required for bias mitigation, from foundational big data systems to advanced cognitive computing platforms. Each sub-branch includes reference ranges demonstrating the empirical basis for these technological capabilities.

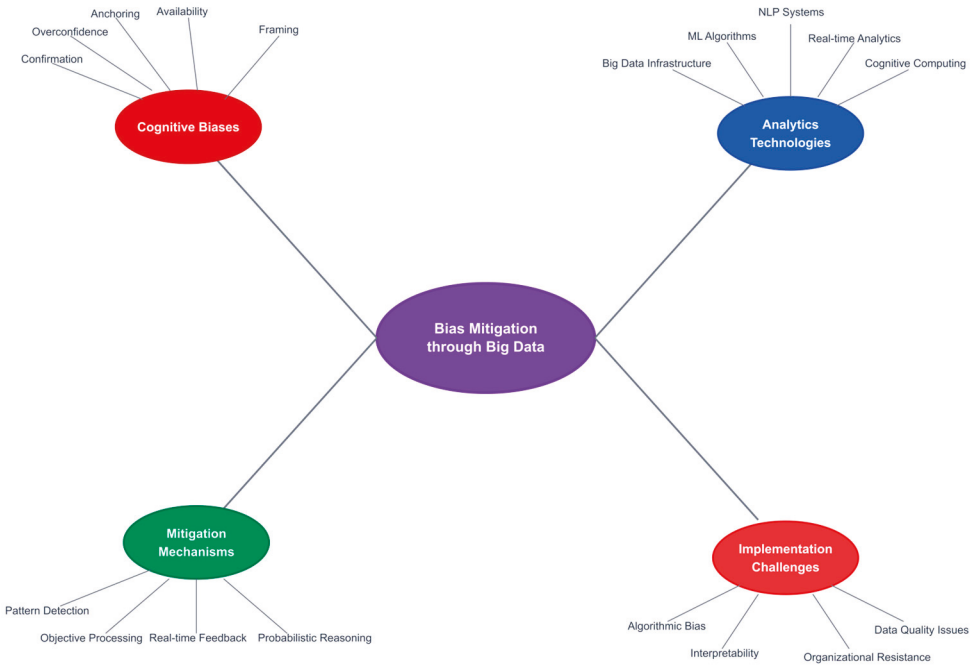


Figure 5. Key concepts mind map in bias mitigation through big data analytics.

The Mitigation Mechanisms branch (green, lower left) explicates the four primary pathways through which analytics addresses cognitive biases: comprehensive pattern detection that reveals hidden relationships, objective processing free from emotional influence, real-time feedback enabling rapid bias recognition, and probabilistic reasoning that counters overconfidence. The Implementation Challenges branch (red, lower right) acknowledges critical barriers including algorithmic bias risks, interpretability requirements, organizational resistance, and data quality concerns.

The radial structure emphasizes that effective bias mitigation requires simultaneous attention to all four domains. The connecting lines represent the dynamic interactions between domains—for instance, how organizational resistance (a challenge) can undermine the deployment of analytics technologies, or how specific mitigation mechanisms directly address particular cognitive biases. This holistic view underscores that technological solutions alone are insufficient; successful bias mitigation demands integrated approaches addressing psychological, technological, methodological, and organizational dimensions simultaneously.

3. Cognitive Biases in Strategic Decision-Making

Cognitive biases systematically impede effective executive decision-making, affecting strategic choices and investment allocations before their consequences become apparent. Despite access to unprecedented data volumes, modern organizations frequently base strategic decisions on mental shortcuts and systematic judgment distortions. This section examines how cognitive biases manifest in executive environments, their effects on business strategy formulation, and the substantial challenges organizations face in detecting and mitigating these biases [112].

3.1. Challenges in Identifying and Mitigating Biases in Executive Contexts

Research consistently demonstrates cognitive biases' pervasive influence on executive decision-making, yet these biases prove extraordinarily difficult to identify and correct in live operational contexts [113]. Strategic decisions present unique challenges due to their

long-term implications, multifaceted external influences, and inherent unpredictability, making it exceptionally difficult to distinguish genuine strategic errors from natural risks associated with operating in uncertain business environments [114].

Table 3 summarizes five fundamental barriers to detecting and managing biases within organizational decision processes:

Table 3. The main challenges in identifying and mitigating biases.

Challenge	Key Issue	Explanation
Bias Blind Spot & Self-Attribution	Failure to recognize personal biases	Executives easily identify biases in others but fail to acknowledge their own, reducing self-correction
Organizational Culture & Groupthink	Suppression of dissent	Dominant leadership and consensus-driven cultures reinforce biases rather than challenge them
Ambiguous & Delayed Feedback	Difficulty learning from past mistakes	Strategic decisions unfold over long timeframes, making it hard to isolate the role of bias in outcomes
Complexity of Attribution	Misdiagnosis of failure	Executives may blame external factors instead of recognizing cognitive distortions in their decision-making
Lack of Structured Processes	Absence of systematic bias mitigation	Unlike other disciplines, corporate strategy lacks formal mechanisms (e.g., checklists, red-team reviews)

The "bias blind spot" represents the most fundamental challenge—individuals' tendency to recognize cognitive distortions in others while remaining oblivious to identical patterns in their own thinking [115]. The combination of successful career progression and industry experience frequently leads executives to develop profound confidence in their judgment, substantially reducing receptiveness to the possibility of cognitive distortion affecting their reasoning processes. Self-attribution bias compounds this problem, as executives typically attribute failures to external market conditions rather than evaluating their own cognitive blind spots [116].

Organizational culture and groupthink dynamics further complicate bias identification at executive levels. High-stakes corporate decisions typically occur within executive teams characterized by dominant personalities and hierarchical structures that implicitly discourage dissent. Without mechanisms actively promoting cognitive diversity, structured red-team analysis, and formalized devil's advocacy, executive teams develop tunnel vision that renders them oblivious to their cognitive distortions [117].

The feedback mechanisms within executive decision processes operate through fundamentally ambiguous and delayed pathways. Strategic decisions typically unfold over extended timeframes during which multiple external factors influence outcomes [118]. This complex causal environment makes it extraordinarily difficult to isolate which specific cognitive biases contributed to strategy failure. Research scholars face similar challenges measuring cognitive bias effects on firm performance, as isolating specific biases from complex organizational outcomes presents significant methodological difficulties [119].

Unlike fields such as medicine and aviation that employ structured error-reduction protocols, executive decision-making has historically lacked formal debiasing procedures. While some organizations have begun implementing debiasing interventions, including

pre-mortem exercises, cognitive bias checklists, and AI-driven decision-support tools, formal bias detection and correction mechanisms remain rare [120].

3.2. Real-World Examples of Executive Biases

Historical corporate records reveal numerous strategic missteps and missed innovation opportunities attributable to cognitive distortions. These cases provide concrete evidence of how biases operate in practice and their substantial organizational costs [121].

3.2.1. Hindsight Bias—The Case of Missed Innovations

Hindsight bias manifests as the "I-knew-it-all-along" effect, causing individuals to retrospectively perceive events as more predictable than they appeared during their initial occurrence. The failure of established companies to capitalize on photocopying technology represents a well-documented example [122].

During the mid-20th century, Haloid Company (later Xerox) developed revolutionary photocopying technology. Industry leaders IBM and Kodak declined opportunities to invest in or acquire this technology, dismissing its potential as either insufficiently profitable or peripheral to their core operations. Contemporary observers often express bewilderment regarding these companies' failure to recognize the breakthrough potential. However, the decision context was considerably more complex, as prevailing mental models and cognitive biases constrained executives' technological evaluations toward innovations aligned with established business operations [123].

This situation transcended mere negligence, as confirmation bias significantly amplified hindsight effects. IBM and Kodak executives demonstrated strong confidence in their existing product portfolios, selectively processing information that reinforced current strategies while systematically overlooking disruptive alternatives [124]. The combination of selective attention to established business models and resistance to challenging fundamental assumptions resulted in declining an innovation that transformed the office equipment industry.

3.2.2. Representativeness Bias—Misjudging Market Patterns

Representativeness bias involves assuming that prior occurrences and familiar patterns will accurately predict future developments despite fundamental environmental changes [122]. Montgomery Ward's experience following World War II demonstrates this phenomenon's strategic impact.

Montgomery Ward's leadership predicted post-World War II economic depression, assuming conditions would mirror the aftermath of World War I. Based on this perspective, the company implemented an extremely conservative business strategy, avoiding expansion and maintaining defensive operational postures. By contrast, competitor Sears recognized fundamental differences between the two post-war economic environments and pursued aggressive expansion, correctly anticipating consumer spending growth [124].

Montgomery Ward committed a fundamental strategic error by treating historical events as definitive guidance while disregarding substantial economic and social transformations. The United States experienced severe economic depression following World War I, but the post-World War II era presented an entirely different landscape characterized by rapid consumer demand, industrial expansion, and government-supported economic growth [125,126]. This misjudgment produced irreversible consequences as Sears leveraged the economic boom while Montgomery Ward permanently surrendered its competitive standing [127,128].

3.2.3. Overconfidence and Hubris—Value-Destroying Acquisitions

Overconfidence bias creates significant challenges during mergers and acquisitions, as executives frequently maintain unrealistic beliefs regarding their ability to extract value from large transactions [129]. Behavioral finance research consistently demonstrates that overconfident executives typically overpay for acquisitions while underestimating integration challenges [130,131].

The 2001 AOL–Time Warner merger exemplifies how overconfidence bias produces catastrophic acquisition failures. AOL pursued a $165 billion acquisition based on executive conviction that the combined entity would dominate emerging digital media markets [132]. However, fundamental assumptions regarding growth trajectory and business synergies proved fatally flawed. The merger culminated in a historic $99 billion write-off—the largest corporate impairment in history—before complete dismantlement within a decade [133].

This pattern continues in contemporary technology acquisitions where executives project past accomplishments to overestimate their capacity to integrate and expand new business operations [134]. Recent artificial intelligence and fintech acquisitions demonstrate identical patterns, with leadership teams avoiding comprehensive due diligence because intense commitment to success narratives leads them to overlook warning signs [135]. Figure 6 synthesizes the three historical cases examined above, illustrating how different cognitive biases manifest in executive decision-making through distinct yet interconnected patterns.

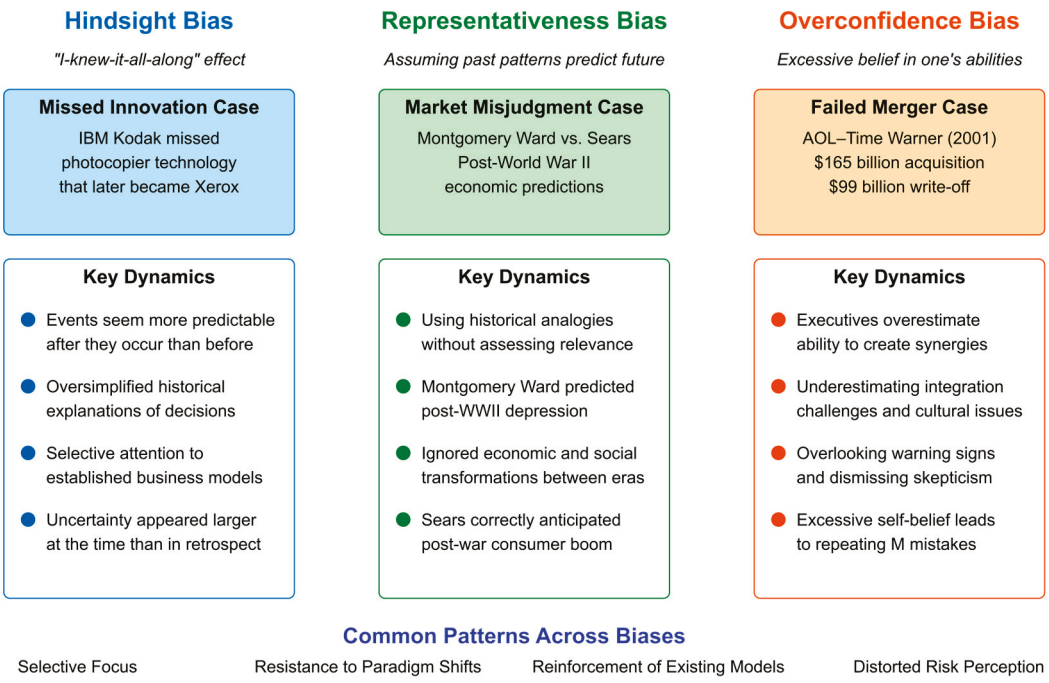


Figure 6. Real-world Examples of Executive Biases in Strategic Decision-Making.

Each bias demonstrates specific dynamics while contributing to four common patterns that transcend individual biases: selective focus, resistance to paradigm shifts, reinforcement of existing models, and distorted risk perception.

The figure reveals how cognitive distortions systematically undermine strategic judgment across diverse business contexts. The hindsight bias panel illustrates how IBM and Kodak's failure to recognize photocopying technology's revolutionary potential only appears "obvious" in retrospect, while uncertainty was substantially greater during the actual decision moment [122–124]. The representativeness bias panel demonstrates Montgomery Ward's critical error in assuming post-WWII conditions would mirror post-WWI

patterns, contrasting sharply with Sears' accurate assessment of the transformed economic landscape [124–127]. The overconfidence bias panel captures how executive hubris in the AOL-Time Warner merger led to catastrophic value destruction through systematic underestimation of integration challenges [132,133].

These patterns—selective focus on confirming information, resistance to paradigm shifts that challenge existing models, reinforcement of established frameworks, and systematically distorted risk perception—represent fundamental cognitive vulnerabilities that persist across different bias types and business contexts. Understanding these common patterns is essential for developing effective bias mitigation strategies through data-driven approaches.

3.3. Implications for Data-Driven Decision Making

The historical cases examined reveal four recurring patterns across all cognitive biases: selective information processing, resistance to paradigm shifts, reinforcement of existing models, and distorted risk perception. These patterns demonstrate why traditional intuition-based decision-making proves inadequate for modern business complexity [136].

3.3.1. Why Analytics Matter

Each examined failure—IBM/Kodak, Montgomery Ward, and AOL–Time Warner—occurred despite available data that could have prevented strategic missteps. This paradox highlights the critical gap between having information and processing it objectively. Modern organizations face exponentially greater complexity than historical counterparts, creating decision contexts where human cognitive limitations become increasingly problematic [137].

Big data analytics offers specific countermeasures to each bias pattern:

- Against Selective Processing: Analytics examines all data without predetermined filters, revealing patterns that confirmation bias would obscure
- Against Paradigm Rigidity: Predictive models generate future scenarios based on emerging trends rather than historical analogies
- Against Echo Chambers: Data democratization enables evidence-based challenges to senior assumptions
- Against Risk Distortion: Probabilistic modeling quantifies uncertainty, replacing subjective optimism with empirical ranges

3.3.2. Implementation Requirements

Analytical tools alone cannot mitigate biases without corresponding organizational changes:

1. Cultural Shift: Organizations must value empirical evidence over hierarchical opinion, replacing "HIPPO" (Highest-Paid Person's Opinion) dynamics with data-driven cultures [136]
2. Process Integration: Analytics must be embedded throughout decision cycles, not treated as optional validation
3. Executive Literacy: Leaders need sufficient analytical understanding to interpret insights appropriately
4. Clear Governance: Frameworks must define when analytical insights override intuition and when human judgment remains primary [137]

3.3.3. Toward Augmented Decision-Making

The objective is augmenting rather than replacing human judgment. Humans excel at contextual understanding and ethical reasoning; analytics provides objective pattern

recognition and bias detection. This symbiosis creates decision-making capabilities exceeding either approach alone. The following sections examine how big data analytics and AI systems operationalize these bias mitigation principles, transforming theoretical frameworks into practical decision enhancement tools.

4. Big Data Analytics as a Tool for Bias Mitigation

Big data analytics offers considerable potential to counteract cognitive biases in executive decision-making by supplanting subjective intuition with empirical, data-driven insights, enabling leaders to make more objective, evidence-based strategic choices. Cognitive biases, including confirmation bias, overconfidence, anchoring, and hindsight bias, frequently distort perception and risk assessment, resulting in suboptimal business decisions, inefficient resource allocation, and missed opportunities [138]. Through leveraging advanced artificial intelligence and machine learning algorithms, statistical modeling techniques, and real-time data processing capabilities, big data analytics can identify hidden patterns, challenge entrenched assumptions, and provide decision-makers with probabilistic rather than intuitive assessments of risks and opportunities. This section explores how analytics mitigates cognitive biases, examines AI's role in identifying and correcting distorted judgment patterns, and discusses structured frameworks that enhance decision quality by integrating big data into executive workflows. We analyze how descriptive, predictive, and prescriptive analytics collectively enhance objectivity, enabling executives to identify trends, model future scenarios, and receive data-driven recommendations that minimize reliance on cognitive shortcuts [139]. Additionally, we address ethical considerations and limitations of AI-driven bias mitigation, including data reliability concerns, algorithmic bias risks, and integration challenges within traditionally intuition-driven decision environments. By understanding how big data and AI transform decision-making processes, organizations can reduce susceptibility to biases, improve strategic foresight, and cultivate analytical rigor that enhances long-term business resilience and competitive advantage [140]. Figure 7 presents a heatmap evaluating both the impact of five cognitive biases on decision-making and big data's effectiveness in mitigating each bias on a 1–9 scale.

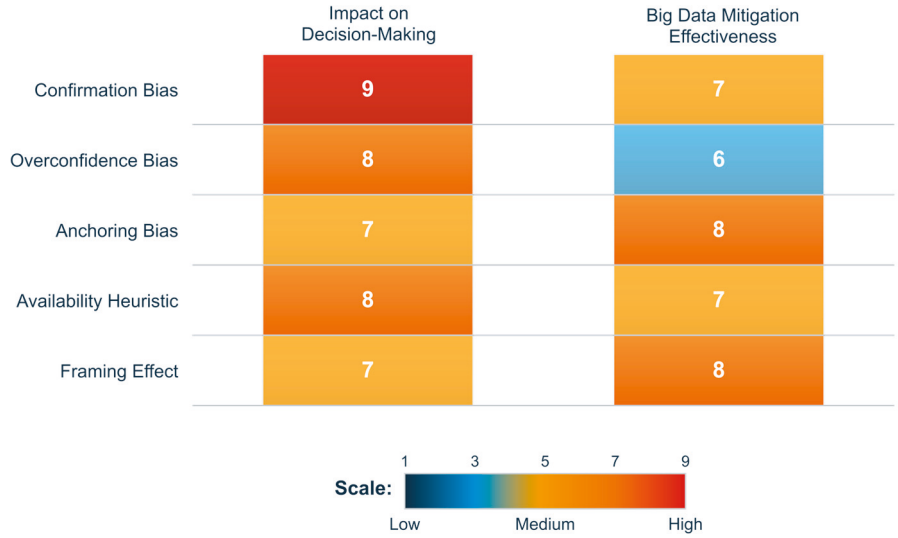


Figure 7. Cognitive biases and Big Data mitigation effectiveness.

The visualization reveals that biases with the highest decision-making impact (confirmation bias: 9) often prove more resistant to mitigation (effectiveness: 7), while biases like

anchoring and framing show stronger mitigation potential (effectiveness: 8) despite lower initial impact (7), highlighting where big data interventions may be most effective.

4.1. Mechanisms for Reducing Bias via Big Data Analytics

Big data analytics is crucial in reducing cognitive bias influence within executive decision-making contexts, functioning as a corrective mechanism that replaces intuition and subjective judgment with empirical, data-driven insights [141]. Cognitive biases frequently emerge because humans rely on heuristics—mental shortcuts that simplify decision-making but introduce systematic errors, particularly within high-stakes, uncertain environments. By contrast, big data analytics processes vast quantities of structured and unstructured information identifies complex patterns and provides objective insights that challenge biased reasoning processes. By offering quantifiable, evidence-based perspectives rather than intuition-driven assumptions, analytics help executives recognize flawed mental models, reevaluate preconceived notions, and make more rational strategic choices [142]. Whether applied to investment decisions, market expansion strategies, product innovation initiatives, or risk assessments, data-driven approaches effectively safeguard against human error, reinforcing analytical rigor throughout executive workflows [143].

4.1.1. Evidence-Based Insights vs. Intuition

Big data analytics significantly reduces cognitive biases by shifting decision-making processes from intuitive judgments toward evidence-based insights derived from comprehensive datasets. Executives traditionally rely heavily on personal experiences or anecdotal evidence, potentially introducing biases such as availability bias, which overemphasizes recent or memorable events. For instance, executives may incorrectly perceive strong organizational performance based solely on recent successes while ignoring objective indicators like market share decline or operational inefficiencies [144,145]. Data analytics challenges these perceptions by providing detailed analyses of market trends, financial metrics, and competitor benchmarks, compelling leaders to confront realities that might contradict their preconceived notions. Another critical bias addressed through big data is confirmation bias, wherein executives selectively accept information aligning with existing strategic beliefs or visions while neglecting contradictory evidence [146,147]. For example, CEOs pursuing market expansion might overlook critical data regarding customer dissatisfaction or regulatory risks, preferentially focusing on supportive evidence. AI-driven analytics mitigates this tendency by presenting comprehensive, unbiased datasets encouraging objective evaluations rather than selective interpretations. Additionally, automated decision-support systems, including predictive analytics platforms and AI-based risk assessment tools, minimize human bias risks affecting the interpretation or manipulation of analytical results [148].

Furthermore, predictive modeling and scenario-based simulations help prevent overconfidence and reliance on simplistic decision heuristics by compelling executives to explore multiple potential future scenarios rather than anchoring exclusively on past successes [149]. For example, AI-powered modeling in mergers and acquisitions generates extensive valuation scenarios that realistically account for economic shifts, competitive dynamics, and integration complexities, thereby avoiding overly optimistic or biased acquisition decisions [150,151]. Similarly, Monte Carlo simulations in corporate risk management provide insights into potential financial, geopolitical, or operational disruptions, supporting comprehensive risk assessment beyond simplistic optimism. Real-time analytics systems also mitigate hindsight bias—the tendency to view outcomes as predictable [152] retroactively. Continuous monitoring and immediate alerts regarding emerging trends, such

as declining revenues or customer churn, facilitate timely corrective actions and prevent post-event rationalizations of strategic missteps [153].

However, big data analytics effectiveness in reducing biases remains contingent upon organizational governance structures, structured integration into decision-making processes, and active bias-awareness training [154]. Without appropriate governance mechanisms, subjective judgments or internal politics may continue overshadowing data-driven insights [155]. Finally, ethical considerations regarding algorithmic biases, data integrity, and interpretability of AI recommendations remain essential, as blindly trusting analytics without critical scrutiny could inadvertently perpetuate biases rather than mitigate them [156].

4.1.2. Comprehensive Pattern Detection

One significant advantage of big data analytics in reducing cognitive biases is its capacity for comprehensive pattern detection at levels far exceeding human cognitive abilities [157]. Regardless of expertise, human executives inherently face limitations in processing extensive, multidimensional datasets, often resulting in misinterpretations, selective attention, and reliance on preconceived notions. By contrast, big data analytics utilizes advanced machine learning algorithms and statistical methods to uncover complex hidden patterns and relationships within vast datasets. These analytical models introduce objectivity into strategic decision processes by systematically analyzing large-scale data across multiple variables and extended time horizons. This approach proves especially effective against confirmation bias, anchoring bias, and representativeness bias, ensuring decisions remain grounded in the complete range of available data rather than selectively filtered subsets [158]. Specifically, big data-driven pattern detection combats confirmation bias through automated hypothesis testing and anomaly detection. Traditionally, executives begin with preconceived assumptions based on personal experiences or anecdotal evidence, selectively seeking data supporting these beliefs while disregarding conflicting information [159]. Big data models avoid these limitations by objectively evaluating relevant data and identifying correlations, trends, and outliers that challenge conventional wisdom. For instance, while retail executives might presume specific customer segments unprofitable based on limited prior assumptions, advanced analytics could reveal hidden profitability through deeper, granular insights into cross-category spending, seasonal behaviors, or cost dynamics [160–162]. These analytical capabilities offer critical reality checks, compelling executives to reevaluate outdated assumptions and enhancing overall strategic accuracy [163].

Additionally, big data analytics provides insights into counterintuitive correlations and complex cause-effect relationships often overlooked by heuristic-driven executive decision-making. Executives traditionally rely on simplified mental models and rules of thumb that inadequately capture the complexity of consumer behavior, financial markets, and technological adoption trends [164]. Machine learning algorithms can examine numerous interactions across diverse data streams, discovering unexpected relationships that challenge established industry norms [165]. For example, detailed data analysis might reveal that extensive product discounting attracts primarily transient, price-sensitive buyers rather than fostering lasting loyalty, thus contradicting prevailing business strategies. Such findings encourage executives to refine strategic approaches based on nuanced, real-world behavioral insights instead of oversimplified assumptions [166].

Moreover, big data analytics' real-time adaptive capabilities significantly mitigate biases associated with historical anchoring and inertia-driven decision-making [167]. Executives frequently rely on outdated strategies rooted in past experiences or static assumptions that no longer reflect current market dynamics. Advanced real-time processing

and streaming analytics continuously update organizational insights, allowing executives to rapidly adapt to evolving conditions such as technological disruptions, shifting consumer preferences, and competitive threats [168]. Big data analytics also addresses cognitive distortions at organizational levels, particularly combating groupthink in collective decision-making environments. It functions as an unbiased arbitrator, providing objective, empirical evidence that reduces reliance on hierarchical dynamics and prevailing consensus. For instance, when executive committees strongly favor particular strategic initiatives, analytics can independently evaluate feasibility and expected outcomes, ensuring decisions incorporate rigorous empirical analysis rather than simply reinforcing internal group beliefs [169,170].

4.1.3. Filtering Out Irrelevancies

A fundamental challenge in human decision-making involves susceptibility to saliency bias—the tendency to disproportionately emphasize highly noticeable yet irrelevant information while undervaluing subtle but statistically significant variables [171]. This bias appears particularly pronounced in executive decision-making, recruitment processes, financial forecasting, and risk management contexts, frequently causing suboptimal judgments and outcomes. AI-driven analytics effectively counteracts this bias by systematically isolating critical predictive factors from vast datasets while disregarding irrelevant, emotionally charged, or attention-grabbing elements [172]. Unlike human decision-makers influenced by experience, narratives, cognitive shortcuts, or external pressures, AI models objectively evaluate data based solely on quantifiable predictive relevance. For instance, in recruitment processes, AI-powered hiring algorithms can eliminate unconscious demographic biases by ignoring factors such as names, gender, ethnicity, or appearance, focusing exclusively on objective criteria such as skills, qualifications, and past performance metrics. This approach ensures hiring decisions remain merit-based, fair, and aligned with genuine talent requirements, significantly improving organizational equity and efficiency [173–175].

In financial decision-making and investment analysis, saliency bias often emerges when human investors react excessively to short-term market volatility, sensational news, or emotional triggers, neglecting long-term fundamentals and statistically validated indicators [176]. AI-driven investment models combat this bias by employing deep learning, time-series forecasting, and sentiment analysis to identify patterns based on meaningful data rather than superficial market fluctuations. These advanced analytical methods help investors avoid reactionary decisions, such as panic-selling during financial crises, by providing robust, objective insights into long-term asset performance and economic trends [177]. Similarly, AI-based analytics significantly enhance corporate risk assessment and fraud detection by eliminating subjective interpretations often influenced by intuition or historical biases. AI platforms systematically filter large datasets, objectively identifying genuine risk signals without biases related to demographic backgrounds, institutional affiliations, or outdated assumptions [178]. For example, AI-powered credit scoring models prioritize behavioral indicators such as repayment habits and spending behaviors rather than irrelevant demographic factors or income brackets. This approach not only increases fairness and financial inclusion but also ensures more accurate and risk-sensitive lending decisions [179,180].

Furthermore, in corporate strategic planning, AI-driven analytics prevent executives from anchoring decisions to outdated or irrelevant benchmarks and past successes. Traditional strategic planning frequently suffers from reliance on historical metrics or experiences that may no longer reflect current market realities, competitive dynamics, or technological disruptions. AI-driven systems mitigate these risks by dynamically evaluating recent consumer behaviors, macroeconomic conditions, and competitive pressures specific to current

strategic contexts, thus promoting adaptability and precision in decision-making [181–183]. Ultimately, AI's capability to filter out irrelevant information represents not merely an incremental efficiency improvement but a fundamental shift toward more rational, data-grounded organizational decisions across multiple business domains [184]. However, this powerful potential must be complemented by rigorous oversight, continuous model validation, and ethical governance frameworks, as AI itself can inadvertently propagate biases if trained on biased datasets or selectively overridden by human judgment. Companies effectively integrating these bias-filtering AI mechanisms will thus achieve significant competitive advantages by consistently making rational, accurate, and strategically sound decisions in complex, rapidly evolving business environments [185,186].

4.1.4. Consistency and Repetition

A critical limitation inherent in human decision-making processes is their fundamental inconsistency. Executive and professional judgments frequently vary due to emotional fluctuations, fatigue, cognitive load, external pressures, and other contextual influences [187]. Even seasoned decision-makers may reach divergent conclusions under identical conditions, introducing elements of randomness, emotional bias, and unpredictability into strategic choices. AI-driven decision models effectively overcome these limitations by consistently applying logical frameworks and systematic reasoning processes, remaining unaffected by emotional states or mental fatigue [188,189]. Unlike human decision-makers, AI systems ensure that identical inputs reliably produce similar outputs, significantly reducing variability and arbitrary biases in decision processes.

This consistency proves particularly advantageous in corporate strategy formulation and financial forecasting contexts. For instance, executives reviewing investment opportunities may shift their stance based on transient mood states or temporary market fluctuations rather than stable, objective evaluation criteria. AI-powered financial modeling and forecasting systems eliminate inconsistencies, ensuring decisions reflect empirical data and probabilistic analysis rather than subjective sentiments or emotional reactions [190,191]. Consequently, organizations implementing AI-driven decision-support tools achieve more accurate, reliable, and rational long-term strategic planning processes that demonstrate reduced vulnerability to emotional or transient influences [192].

Human inconsistency significantly impacts hiring decisions and talent management functions, as recruitment professionals often inadvertently apply varying evaluation standards to similar candidates due to factors including fatigue, cognitive overload, or irrelevant contextual elements [193]. Behavioral research consistently demonstrates that candidate assessments frequently differ based on interview timing, preceding candidate quality, or evaluators' momentary psychological states. AI-driven hiring algorithms eliminate this subjective variability by strictly adhering to predefined selection criteria and consistently evaluating candidate attributes, including experience, skill levels, and demonstrated performance metrics [194]. For example, an AI model ranking candidates based on objective data will produce identical outcomes regardless of contextual influences or evaluator biases. This standardization enhances fairness, predictability, and efficiency in recruitment processes, minimizing subconscious biases and significantly improving talent acquisition objectivity.

Similarly, clinical decision-making and healthcare diagnostics frequently suffer from human judgment variability attributable to cognitive fatigue, workload stress, and patient-specific contextual factors. Research studies highlight that clinicians' medical decisions can differ significantly depending on evaluation timing—such as assessments conducted at the end of tiring shifts or following complex patient cases [195]. AI-driven diagnostic tools and clinical decision-support systems eliminate these inconsistencies by applying standardized evaluation protocols based strictly on symptomatology, medical history,

and validated statistical models. Technologies, including AI-based radiology diagnostics, disease-risk prediction algorithms, and AI-guided treatment recommendation systems, ensure uniformity and accuracy regardless of external contextual or emotional factors (Figure 8). This systematic approach significantly reduces diagnostic errors and enhances patient outcomes by guaranteeing consistent treatment protocols based solely on objective clinical evidence [196].

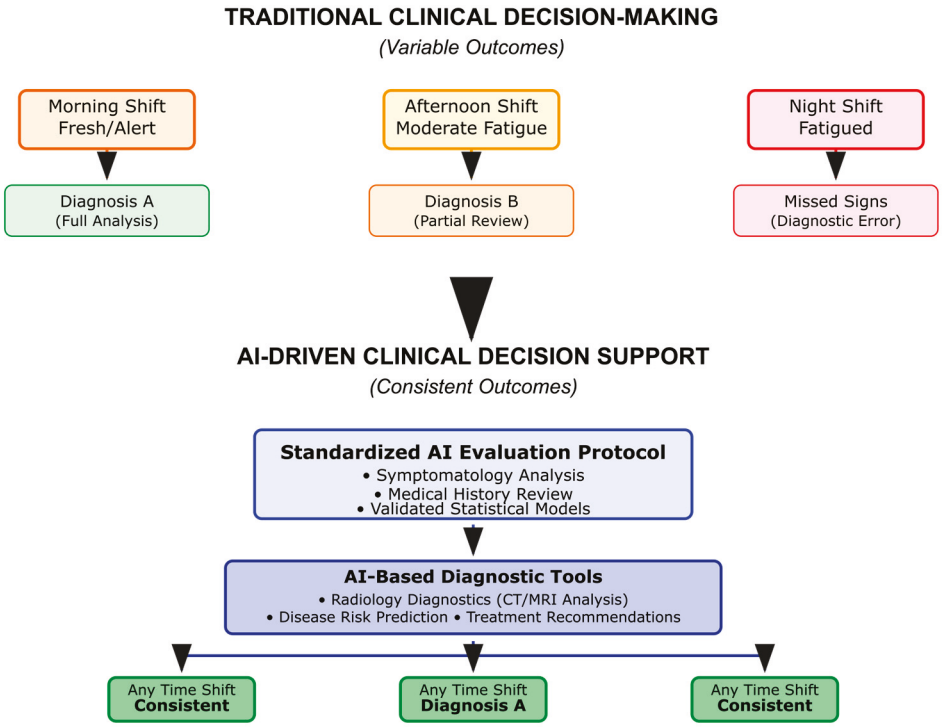


Figure 8. AI-driven consistency in clinical decision-making.

From corporate governance and leadership perspectives, integrating AI-driven consistency into decision-making frameworks supports sustained organizational stability and strategic coherence [197]. Executives frequently shift strategies impulsively in response to transient events, media narratives, or short-term market volatility, resulting in erratic strategic trajectories and organizational inefficiencies. AI analytics platforms address this instability by providing structured, algorithmically validated analyses firmly grounded in historical patterns, probabilistic forecasting, and comprehensive risk assessment methodologies [198]. Companies adopting AI-enhanced scenario simulations and strategic planning tools thereby avoid reactionary and short-term thinking patterns, ensuring steady execution of data-informed, long-term strategies and securing sustained competitive advantages in increasingly dynamic business environments [199].

4.2. Role of AI and Machine Learning in Detecting and Countering Bias

Artificial intelligence (AI) and machine learning (ML) have revolutionized bias detection and mitigation within decision-making processes, particularly in high-stakes environments where human cognitive limitations and systemic biases have historically influenced outcomes. Whether unconscious or structural, bias often remains undetected due to deep integration within established business practices, decision frameworks, and historical datasets [200]. AI and ML technologies have introduced data-driven methodologies enabling organizations to identify, quantify, and mitigate bias in previously impossible ways. One of ML's most impactful applications in bias detection involves its capacity to analyze past decisions and decision-making processes at scale, statistically identifying patterns

indicating bias presence [201]. By leveraging pattern recognition techniques, anomaly detection algorithms, and fairness-aware computational frameworks, AI systems can scrutinize historical data, detect systematic disparities, and provide quantifiable evidence of biased trends. This capability is particularly valuable in corporate hiring, financial lending, law enforcement, healthcare diagnostics, and algorithmic governance, where unchecked human bias can lead to inequitable, inefficient, or legally problematic decisions [202].

4.2.1. Bias Detection—Identifying and Correcting Systemic Inequities in Decision-Making Through ML and AI

Machine learning significantly enhances bias detection capabilities by analyzing historical decision-making data to reveal correlations with irrelevant or ethically problematic factors. ML algorithms examine historical lending data in financial contexts to identify hidden demographic biases, such as race or gender, influencing loan approvals despite identical financial qualifications [203]. Similarly, within corporate environments, AI analytics examine hiring and promotion records to pinpoint statistical patterns potentially signifying systemic biases based on characteristics including gender, age, or ethnicity that cannot be justified solely through objective performance measures [204]. By quantifying these biases clearly and objectively, AI enables organizations to proactively correct discriminatory practices, ensuring decisions align strictly with merit-based criteria [205]. ML algorithms also proactively detect biases by flagging anomalies in real-time decision-making processes. Unlike traditional human judgments susceptible to cognitive shortcuts or subjective biases, AI systems can implement immediate real-time anomaly detection, highlighting deviations from fair or optimal decisions. For instance, in judicial contexts, AI tools can monitor court decisions to detect and address disparities promptly, significantly reducing biases influenced by race, judge fatigue, or personal heuristics [206]. Similarly, in healthcare settings, AI-driven diagnostics ensure treatment recommendations based strictly on medical data rather than demographic characteristics historically linked to biased medical decisions, thereby correcting disparities in treatment and diagnosis among marginalized populations [207,208].

However, ML systems are not inherently bias-free; their effectiveness depends heavily on training data quality and neutrality. Historical biases embedded in datasets related to hiring, lending, or criminal sentencing may inadvertently transfer into AI algorithms, perpetuating rather than eliminating discrimination [209]. Therefore, rigorous ethical reviews, fair auditing, and algorithmic transparency are crucial safeguards. Techniques including adversarial debiasing, algorithmic auditing, and synthetic data augmentation are actively pursued within fairness-aware machine learning research to prevent AI systems from replicating or exacerbating existing biases [210]. Consequently, as organizations integrate AI more deeply into decision-making processes, embedding robust fairness measures, continuous bias detection mechanisms, and ethical oversight frameworks becomes essential for AI to function effectively as a corrective mechanism against systemic inequities rather than inadvertently reinforcing them [211].

4.2.2. AI-Driven Decision Support Systems—Counteracting Human Biases Through Algorithmic Guidance

AI-driven decision support systems (DSSs) have emerged as critical tools in corporate strategy, financial analysis, and policy-making, offering data-backed alternatives to intuition-based human decisions while mitigating cognitive biases, including confirmation, availability, and anchoring biases. Unlike traditional analytical tools, AI-based DSSs transcend merely providing information—they actively participate in decision-making by generating predictive insights, modeling alternative scenarios, and systematically challenging entrenched assumptions through computational reasoning [212]. These systems

reduce bias by presenting decision-makers with alternative scenarios and outcomes, directly countering tendencies toward tunnel vision or self-reinforcing cognitive distortions. For instance, in strategic planning contexts, AI-powered tools can simulate multiple market entry scenarios, including options initially disregarded by executives due to personal biases or preconceived notions about consumer behavior or competition. By grounding these scenarios in robust predictive analytics, economic modeling, and consumer sentiment analysis, AI compels decision-makers to adopt broader, evidence-based perspectives rather than relying on familiar yet biased heuristics [213–215].

AI-driven DSSs are particularly influential in high-stakes financial environments, exemplified by hedge funds and algorithmic trading firms prioritizing data-driven logic over emotional or hierarchical biases [216]. Bridgewater Associates, a leading hedge fund, utilizes algorithmic systems to ensure investment decisions reflect empirical credibility and factual performance data ("believability") rather than executive seniority or organizational influence, preventing social or hierarchical biases from distorting investment choices [217]. Similarly, AI financial modeling systems facilitate unbiased portfolio management by forecasting market movements and enabling fund managers to test multiple portfolio strategies rigorously through data-backed simulations. Beyond finance, AI decision-support tools significantly enhance corporate governance and organizational decision-making by addressing personal favoritism, institutional inertia, and legacy biases. In HR and talent management contexts, AI-based systems objectively assess promotion and resource allocation decisions based on empirical data, including productivity metrics and leadership performance, rather than subjective managerial evaluations [218,219]. Strategic planning is similarly transformed, with AI-driven analytics compelling executives to regularly reconsider outdated business models by continuously integrating competitive analysis, consumer behavior insights, and emerging technology trends, ensuring forward-looking decisions based strictly on objective data [220].

Despite these advantages, AI-driven DSSs also present inherent limitations, as AI algorithms can perpetuate biases if trained on biased historical datasets or incorporating flawed assumptions within their models [221]. For example, an AI hiring model might unintentionally replicate gender biases prevalent in historical hiring data, mistakenly interpreting these patterns as meritocratic standards rather than reflecting systemic inequities. Addressing such issues requires implementing fairness-aware machine learning techniques, adversarial debiasing methods, and rigorous, continuous auditing processes to detect and mitigate embedded biases. Ultimately, AI should function as an augmentative decision-support tool rather than an absolute authority, preserving critical human oversight to ensure AI-driven recommendations receive appropriate scrutiny and ethical consideration [222].

4.2.3. Cognitive Collaboration—AI as a Real-Time Debiasing Partner in Decision-Making

The concept of cognitive collaboration, where AI actively engages with human decision-makers to identify and mitigate biases during real-time strategic deliberations, represents a significant advancement beyond traditional passive AI decision support. Leveraging techniques including natural language processing (NLP), sentiment analysis, and real-time analytics, cognitive AI assistants actively monitor executive discussions, detecting cognitive distortions like groupthink, overconfidence, and confirmation bias, intervening when necessary to prompt balanced, empirically informed considerations [223,224]. These interactive systems do not replace human intuition; they function as proactive corrective mechanisms, ensuring strategic deliberations remain grounded in rational analysis and comprehensive evaluation rather than subjective biases.

AI-driven cognitive collaboration proves particularly effective in executive contexts, including strategic planning meetings, investment discussions, and boardroom decision-

making, where bias-driven narratives or reliance on past successes often dominate discussions. AI cognitive assistants function as real-time "debiasing coaches," analyzing ongoing conversations to identify blind spots and biases, then proactively prompting participants to consider overlooked risks or alternative perspectives. For instance, during market expansion planning, if discussions disproportionately emphasize potential gains without adequately addressing risks or competitive challenges, AI can highlight this imbalance, recommending historical precedent examination and risk factor analysis. Similarly, when executives anchor too heavily on past successes (representativeness bias), AI interventions encourage independent assessments of current market conditions, ensuring strategies remain data-driven rather than emotionally influenced [225–227].

Integrating cognitive AI assistants aligns with augmented intelligence principles, where human agency remains central while AI primarily eliminates cognitive blind spots. This symbiotic approach significantly enhances decision-making in high-stakes sectors, including finance, healthcare, and policy-making, where bias can produce serious long-term consequences. For example, in financial contexts, AI can flag excessive focus on short-term market fluctuations, redirecting fund managers toward long-term macroeconomic indicators; similarly, in clinical environments, AI can monitor physician discussions to ensure treatments are statistically justified, free from biases associated with patient demographics [228,229]. As AI capabilities in conversational analytics and predictive modeling continue evolving, cognitive collaboration promises to transform executive decisions from intuition-based to consistently bias-aware, data-informed deliberations [230].

Although AI-driven cognitive collaboration remains in the early development stages, its potential to proactively identify biases, foster rational discussions, and promote evidence-based decision-making is immense. Organizations adopting these advanced AI tools gain the capacity to address cognitive biases in real time rather than post-hoc, significantly enhancing decision quality [231]. However, effectively leveraging cognitive collaboration requires cultivating receptive organizational cultures that embrace AI-driven interventions, prioritizing objective data over personal convictions. Companies successfully integrating cognitive AI into decision-making processes mitigate biases and position themselves strategically to achieve sustained competitive advantages in increasingly complex, data-centric environments [232].

4.2.4. Objective Optimization—AI-Driven Decision Models as a Benchmark for Unbiased Strategy

Machine learning significantly transforms executive decision-making by optimizing outcomes objectively, independent of subjective human intuition or organizational hierarchy. Traditional corporate decision processes, frequently influenced by hierarchical biases such as the HIPPO effect (highest-paid person's opinion), often result in suboptimal resource allocation and strategic miscalculations driven more by seniority and personal conviction than analytical rigor [233]. ML models address this issue by evaluating strategic alternatives based on empirical efficiency and outcome optimization, effectively insulating decisions from emotional, reputational, or institutional biases [234].

Marketing budget allocation provides an illustrative example, where traditional approaches typically favor historical practices, executive intuition, or legacy marketing channels rather than real-time, data-driven ROI analysis [235]. ML-driven models analyzing consumer engagement and market responsiveness may recommend shifting resources toward modern digital channels or influencer marketing, often contradicting traditional executive preferences rooted in brand strength or past successes [236]. When executives resist AI-driven recommendations without clear empirical justification, this discrepancy highlights underlying cognitive biases, such as anchoring or status quo bias [237]. Consequently, organizations using AI-driven budget allocations as benchmarks can systematically identify

and challenge bias-driven inefficiencies, ensuring strategic decisions remain grounded in performance data rather than habitual or hierarchical influences [238].

AI-driven predictive modeling also reshapes financial forecasting and corporate performance management by counteracting optimism or pessimism biases, such as overconfidence or loss aversion, frequently distorting strategic planning. Executives commonly inflate revenue projections or underestimate operational risks based on wishful thinking or prior successes [239]. AI-generated predictive models provide objective benchmarks for forecasts, compelling managers to systematically justify any deviations. Persistent deviations from these AI-established benchmarks without robust justification typically indicate cognitive biases influencing managerial expectations [240]. Organizations increasingly implement formal decision audits using AI-generated projections to instill disciplined, evidence-based decision-making practices, enhancing forecast accuracy and minimizing emotional or politically influenced distortions [241].

Moreover, AI optimization models profoundly impact broader corporate structures by shifting reliance from intuition-based leadership toward data-driven, algorithmically validated strategies. In sectors including logistics, operations, and dynamic pricing, AI models adjust strategy in real time based solely on evolving data, unaffected by human biases that frequently foster decision inertia or resistance to change [242]. For example, dynamic pricing algorithms in e-commerce establish revenue-maximizing price points that human managers might avoid due to concerns over customer perception or internal politics. Similarly, AI-driven logistics optimization can recommend fundamental shifts in supply routes or warehouse locations that challenge traditional industry practices. When AI-proposed strategies consistently outperform conventional approaches yet human leaders reject them without empirical rationale, biases such as attachment to legacy processes or aversion to algorithmic management become evident [243].

4.3. Empirical and Experimental Methods for Evaluating Bias Reduction

Ensuring that big data analytics and AI-driven interventions genuinely reduce cognitive bias in decision-making requires rigorous empirical validation. While theoretical discussions on machine learning potential, algorithmic transparency, and data-driven decision-making highlight promising possibilities, empirical research and experimental methodologies prove their effectiveness [244]. Without proper empirical validation, organizations risk assuming that AI and big data approaches are inherently unbiased when their impact depends on careful implementation, monitoring, and validation through robust evaluation frameworks [245]. To determine whether bias mitigation strategies lead to measurable improvements in decision quality, fairness, and objectivity, companies and researchers employ various quantitative and experimental methodologies, with A/B testing emerging as one of the most direct and practical approaches for assessing bias reduction technique impacts [246]. Table 4 presents an overview of the four primary empirical and experimental methods for evaluating bias reduction efforts in corporate decision-making.

Figure 9 illustrates the four principal approaches through which artificial intelligence and machine learning mitigate cognitive biases in executive decision-making. At the center, a neural network within a brain symbolizes the integration of AI with human cognition. The four quadrants represent distinct but complementary methodologies: (1) Bias Detection systems that analyze historical data to identify systematic bias patterns in processes like hiring and lending; (2) Decision Support Systems that provide data-based alternatives to intuition through multi-scenario modeling and merit-based assessments; (3) Cognitive Collaboration tools that monitor discussions in real time to counteract emerging biases during strategic planning and boardroom decisions; and (4) Objective Optimization frameworks that establish data-driven benchmarks eliminating hierarchical biases in resource allocation

and pricing. The diagram acknowledges both limitations (inherited biases from training data, explainability challenges) and benefits (decision consistency, reduced emotional influence) of AI-driven bias mitigation, emphasizing that these technologies function as detection mechanisms for existing biases and as proactive tools for creating more objective decision environments.

Table 4. The main 4 empirical/experimental methods for evaluating bias reduction.

Empirical and Experimental Methods	References	Description
A/B Testing of Decision Processes	[247–249]	A/B testing enhances decision accuracy by reducing post-selection bias, improving statistical inference, and fostering ethical, transparent experimentation in online controlled environments.
Simulation Experiments	[250–252]	Simulation-based decision models and AI-assisted analysis improve accuracy by reducing heuristic biases, ensuring better alignment with rational benchmarks in financial risk assessment, healthcare diagnostics, and search behavior modeling.
Pre- and Post-Analytics Analysis	[253–255]	Pre- and post-analytics analysis enhances decision accuracy by systematically comparing historical and post-implementation outcomes. It also demonstrates reductions in cognitive biases such as overconfidence, planning fallacy, and overreliance on intuition in domains like corporate strategy, finance, and healthcare.
Surveys and Behavioral Assessments	[256–258]	Surveys and behavioral assessments help identify prevalent cognitive biases such as overconfidence, confirmation bias, and anchoring. Proper survey design and structured assessments improve bias detection and mitigation, leading to more accurate decision-making insights.

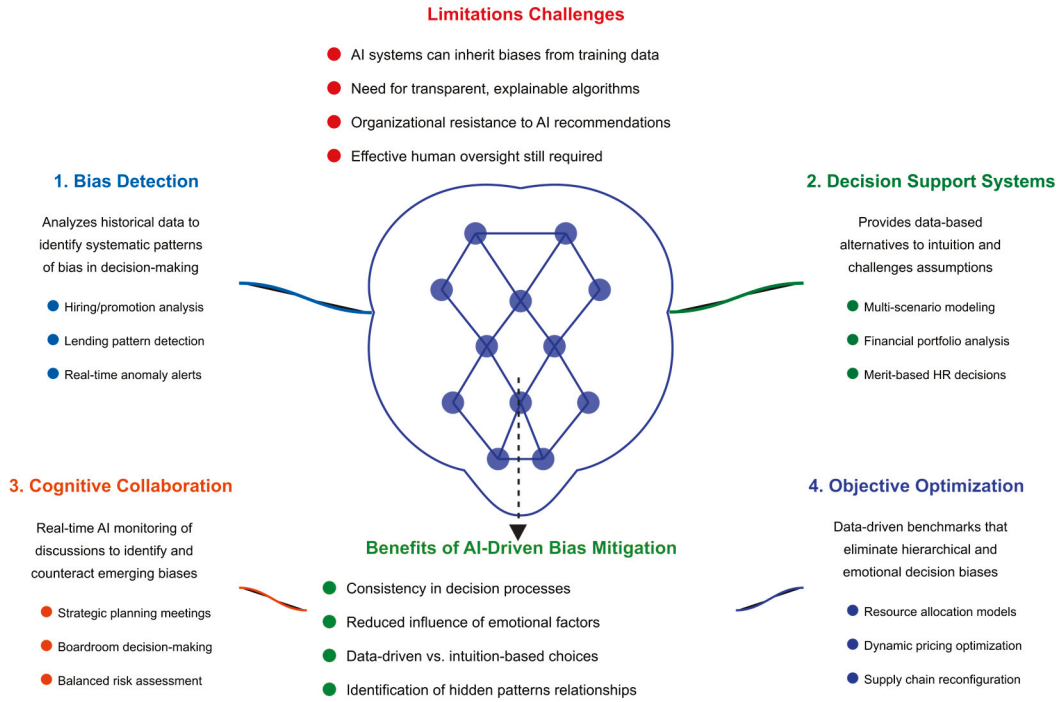


Figure 9. AI and Machine Learning Approaches to Bias Mitigation in Decision-Making.

4.3.1. A/B Testing of Decision Processes

A/B testing, widely implemented across behavioral science, technology, and business analytics domains, represents a robust empirical methodology for evaluating bias-reduction strategy effectiveness by directly comparing human-driven decision-making against AI-augmented approaches [259]. This method isolates the impact of AI-driven tools by dividing decision-makers into two distinct groups—one employing traditional methodologies and the other utilizing AI-assisted systems—to quantitatively assess differences in decision quality, accuracy, and bias reduction [260]. For instance, corporate strategy teams implementing predictive analytics and debiasing algorithms frequently achieve more accurate forecasts and earlier risk detection than intuition-based teams, providing clear empirical evidence of cognitive bias reduction through AI augmentation [261].

This testing approach is particularly valuable in addressing persistent biases in hiring, talent management, and promotion processes. Organizations typically implement these evaluations by assigning one hiring manager group traditional resume screening methodologies. At the same time, another employs AI-powered candidate evaluation systems focused strictly on job-relevant competencies rather than demographic factors [262]. By comparing hiring outcomes, workforce diversity metrics, and employee performance indicators, organizations can empirically determine whether AI-enhanced processes improve fairness, equity, and organizational effectiveness [263]. Financial institutions similarly leverage A/B testing to evaluate AI-driven credit scoring systems, contrasting traditional manual loan approval processes against AI models designed to reduce demographic biases. These institutions compare multiple outcome measures, including approval rates, default probabilities, and customer risk profiles, to ensure AI models effectively address systemic biases without compromising risk assessment accuracy [264,265].

Despite its methodological strengths, A/B testing faces several practical implementation challenges. Defining and quantifying "improved" decisions, particularly regarding bias reduction, often involves complex trade-offs between competing organizational objectives such as operational efficiency, workforce diversity, and profitability [266]. Furthermore, controlling real-world experimental conditions proves difficult due to external variables, including market volatility, consumer behavior shifts, or regulatory changes that may confound test results [267]. Additionally, interpretability and trust in AI-driven decisions remain crucial considerations—organizations must ensure transparency in AI-generated recommendations. When decision-makers perceive AI as a "black box," resistance may significantly diminish its practical effectiveness. Addressing this challenge requires incorporating explainable AI (XAI) methodologies that transparently reveal AI decision-making processes, ensuring accountability, interpretability, and alignment with human judgment frameworks [268].

4.3.2. Simulation Experiments

Simulation experiments provide a rigorous empirical methodology for evaluating artificial intelligence effectiveness in mitigating cognitive biases, particularly within complex decision environments, including finance, healthcare, and strategic planning. Unlike A/B testing approaches, simulation enables controlled decision-making scenarios where optimal decisions or rational benchmarks are clearly defined, facilitating precise quantification of AI's impact on bias reduction, accuracy improvement, and decision consistency [269,270]. For example, investment scenario studies demonstrate that participants utilizing AI-assisted risk analysis tools consistently achieve superior portfolio diversification, reduced susceptibility to biases, including loss aversion and recency effects, and lower decision variance than participants relying solely on human intuition [271,272]. However, AI-driven recommendation effectiveness depends on user trust levels; even highly accurate

AI systems may be disregarded if users remain skeptical or perceive systems as lacking interpretability, highlighting a critical barrier to broader AI adoption [273].

Healthcare simulation studies similarly demonstrate AI's potential to improve diagnostic accuracy and reduce cognitive biases, including availability bias, anchoring, and premature closure. Physicians utilizing AI-supported diagnostic tools trained on comprehensive medical datasets demonstrate higher accuracy rates, faster diagnosis times, and lower variability than colleagues using traditional methods exclusively [274]. However, clinical adoption depends heavily on physicians' willingness to trust and integrate AI recommendations, as clinicians frequently override algorithmic suggestions due to concerns regarding accountability, patient-specific nuances, or perceived transparency limitations [275]. Consequently, AI systems in healthcare contexts must offer accuracy, clear interpretability, and seamless integration with established human clinical reasoning workflows.

Credit risk assessment and lending decisions represent another critical domain where simulation experiments evaluate AI-driven bias mitigation. Historical systemic biases in lending have disadvantaged certain demographic groups due to traditional credit-scoring heuristics. Controlled simulations comparing conventional loan approval practices to AI-driven models—which objectively assess repayment behavior patterns and unbiased risk signals—indicate that AI systems can effectively reduce discriminatory lending patterns while maintaining accurate loan performance prediction [276]. Nevertheless, human biases or resistance to algorithmic decision-making can undermine these improvements when loan officers override AI recommendations due to discomfort with data-driven methodologies [277].

Despite their analytical advantages, simulation experiments face several implementation challenges. Designing realistic experimental tasks that accurately reflect real-world complexities proves essential, as oversimplification risks neglecting critical contextual nuances. Additionally, bias mitigation strategies require careful monitoring to prevent unintended biases from emerging within AI models, particularly when models overemphasize factors that inadvertently disadvantage certain groups [278]. User trust and explainability remain central challenges—decision-makers are unlikely to adopt AI recommendations unless they fully understand and trust the underlying rationale, underscoring the necessity of explainable AI (XAI) techniques. Nevertheless, when carefully designed and implemented, simulation experiments consistently demonstrate that AI-enhanced decision-making frameworks outperform traditional judgment across diverse domains, provided users trust and adhere to algorithmic recommendations [279].

4.3.3. Pre- and Post-Analytics Analysis

Pre- and post-analytics analysis represents a practical methodology for evaluating big data analytics' effectiveness in mitigating cognitive biases, focusing on real-world historical decision-making trends before and after implementing data-driven systems. Unlike controlled experimental approaches such as A/B testing, this method examines actual organizational practices, highlighting systematic improvements in decision rationality, accuracy, and objectivity attributable to analytics integration [280,281]. For instance, in corporate project management contexts, retrospective analyses have demonstrated that organizations frequently suffer from cognitive biases, including overconfidence and optimism, before adopting predictive analytics, resulting in unrealistic projections, cost overruns, and project delays [282]. After implementing analytics-driven forecasting tools utilizing historical data and probabilistic modeling, companies significantly improved project timeline management, resource allocation efficiency, and risk management processes, indicating reduced cognitive bias influence and enhanced strategic accuracy [283,284].

Financial institutions have similarly utilized pre- and post-analytics analysis to demonstrate how big data reduces bias in portfolio management and risk assessment processes. Historically, behavioral biases, including herd mentality, overconfidence, and recency effects, frequently led to irrational portfolio decisions, heightened risk-taking behaviors, and market volatility [285]. Retrospective studies comparing investment decisions before and after introducing AI-driven risk models, predictive stress testing methodologies, and algorithmic trading systems have consistently revealed improved discipline, enhanced risk management, and reduced susceptibility to emotional market reactions [286]. Similar findings emerged within recruitment and human resource management domains, where biases related to subjective heuristics, demographic factors, or personal affinity historically influenced hiring and promotion decisions [287]. Organizations adopting data-driven hiring models—focusing on objective competencies, structured interview evaluations, and predictive analytics—reported increased workforce diversity, improved hiring accuracy, and enhanced employee performance, demonstrating analytics' effectiveness in promoting meritocratic, unbiased talent management [288].

Despite its real-world relevance, pre- and post-analytics analysis faces methodological limitations, particularly difficulty controlling external factors, including market conditions, regulatory changes, or organizational leadership transitions, which may independently influence decision outcomes [289]. Additionally, ensuring historical data reliability and establishing consistent metrics for evaluating decision improvements represent crucial steps for generating actionable insights. Nonetheless, when executed carefully, this method provides compelling longitudinal evidence regarding big data analytics' transformative effects, affirming that data-driven decision-making significantly reduces cognitive biases and enhances decision quality across multiple organizational contexts [290].

4.3.4. Surveys and Behavioral Assessments

Surveys and behavioral assessments complement empirical methodologies by qualitatively capturing executives' perceptions of and responses to big data analytics and bias mitigation tools, providing insights into psychological shifts and cognitive adaptations within decision-making processes [291]. After integrating analytics-driven interventions, surveys offer valuable self-reported data, revealing whether executives perceive reduced influence from subjective factors, including intuition, organizational hierarchy, or peer dynamics.

For example, executives frequently report feeling less susceptible to biases like groupthink or anchoring due to increased reliance on objective, data-supported insights [292–294]. Behavioral assessments complement survey approaches by objectively tracking observable shifts in executive behavior before and after introducing analytics or debiasing interventions [295], evaluating patterns such as:

- Willingness to consider disconfirming evidence: Organizations analyze meeting records to assess whether executives increasingly engage with negative evidence, alternative viewpoints, and AI-generated counter-scenarios, indicating active cognitive broadening and reduced confirmation bias [296].
- Time spent deliberating and weighing trade-offs: Monitoring decision processes can reveal whether structured interventions (e.g., bias checklists or predictive analytics) cause executives to slow down, carefully evaluate trade-offs, and thoroughly consider risks and opportunities, mitigating overconfidence or framing biases [297].
- Shifts in language and argumentation structure: Textual analysis of meeting discussions identifies shifts from categorical, confident language toward more probabilistic and data-referenced reasoning approaches, providing clear evidence of reduced reliance on heuristics and increased integration of analytical perspectives [298].

Longitudinal behavioral assessments further indicate whether executives consistently integrate analytics into their workflows or revert to intuitive decision-making habits, emphasizing the importance of sustaining continuous data-driven practice cultures to avoid bias reemergence [299]. Despite challenges including social desirability bias in surveys and difficulty quantifying behavioral changes, combining qualitative behavioral insights with quantitative performance metrics (e.g., accuracy improvements, forecasting reliability) ensures a comprehensive understanding of how AI-driven analytics reshape executive cognition, organizational decision-making culture, and ultimately decision quality [300].

5. Challenges and Limitations

Despite significant potential, big data analytics faces substantial technical, organizational, and ethical constraints in mitigating cognitive biases in executive decision-making. While AI-driven systems can enhance objectivity and reduce systematic biases, their effectiveness depends critically on data quality, model interpretability, and organizational readiness for adoption [301]. This section examines key implementation barriers that limit the practical application of AI-driven bias mitigation strategies. The comprehensive framework (Figure 10) illustrates the three-layer structure of AI-driven bias mitigation challenges and their corresponding solutions.

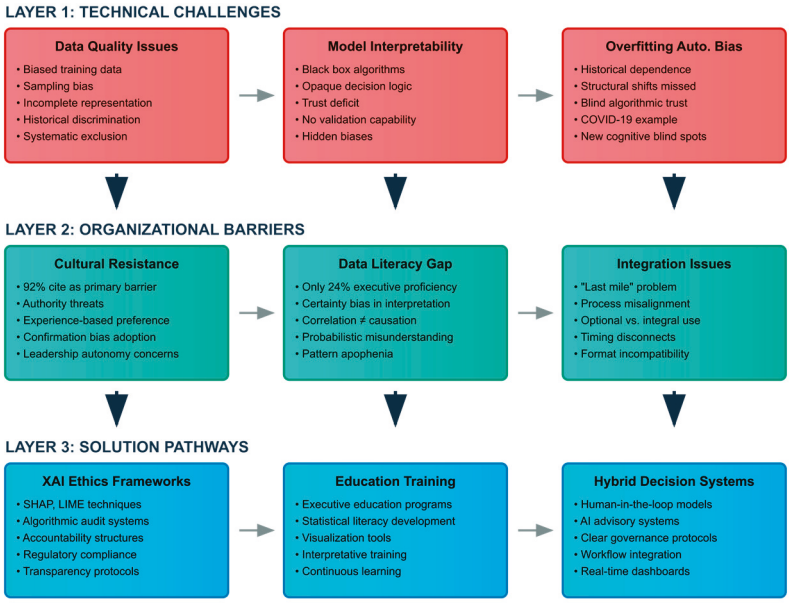


Figure 10. Multi-Dimensional Framework for AI-Driven Bias Mitigation.

The visualization demonstrates how technical challenges at the foundational level cascade through organizational barriers and ultimately require integrated solution pathways. Each layer is interconnected, showing that addressing bias mitigation requires simultaneous attention to technical, organizational, and strategic considerations.

5.1. Technical Limitations of AI-Driven Bias Mitigation

5.1.1. Data Quality and Algorithmic Bias

The fundamental principle of "garbage in, garbage out" critically applies to AI-driven bias mitigation systems. When datasets used to train predictive models contain historical biases, sampling errors, or incomplete representations, the resulting analytics perpetuate rather than eliminate systematic discrimination [302]. This challenge manifests prominently in recruitment and lending decisions, where AI systems trained on historically biased data reproduce discriminatory patterns under the guise of objectivity [303].

Healthcare applications demonstrate particularly severe consequences of biased training data. Medical AI models predominantly trained on data from specific demographic groups fail to provide accurate diagnoses for underrepresented populations, exacerbating rather than reducing healthcare disparities [304]. Similarly, credit scoring algorithms trained on historical lending patterns that systematically excluded low-income or minority borrowers continue reinforcing financial exclusion, even when operating without explicit discriminatory programming [305]. These examples illustrate how algorithmic bias can emerge from training data rather than model design, requiring continuous auditing and diverse dataset curation to ensure representative analytics outcomes.

5.1.2. Model Interpretability and Transparency Challenges

Complex machine learning models, particularly deep neural networks and ensemble methods, often function as "black boxes" that provide accurate predictions without transparent reasoning processes [306]. This opacity creates two critical challenges for executive decision-making: first, leaders cannot understand or validate AI recommendations, leading to mistrust or misapplication of insights; second, embedded algorithmic biases remain undetectable, allowing systematic errors to persist unidentified [307].

Explainable AI (XAI) techniques address interpretability challenges through methods including SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-agnostic Explanations), decision tree visualizations, and feature importance analysis [308]. However, integrating explainability into machine learning systems often requires trade-offs with predictive accuracy, necessitating careful balance between model complexity and interpretability. Without transparency in AI decision-making processes, organizations risk replacing human cognitive biases with algorithmic biases that remain equally undetectable and potentially more systematic in their application.

5.1.3. Overfitting and Automation Bias

A fundamental limitation of historical data-dependent models involves their failure during structural shifts, paradigm changes, or unprecedented events. Models trained exclusively on past patterns may produce misleading predictions when underlying conditions change, a phenomenon known as overfitting to historical data [309]. The COVID-19 pandemic exemplified this limitation when corporate risk models trained on pre-2020 data failed to predict pandemic-related business disruptions, leaving organizations that overrelied on AI-driven risk assessment tools unprepared for unprecedented challenges [310].

This technical limitation connects to automation bias, whereby decision-makers develop excessive trust in algorithmic outputs without maintaining critical evaluation capabilities [311]. When executives blindly trust model-generated insights, they create new cognitive blind spots that may prove more dangerous than the original biases these systems intended to mitigate. Effective implementation requires continuous model validation against evolving real-world conditions and maintaining human judgment as a counterbalance to algorithmic recommendations.

5.1.4. Quantification Limitations in Strategic Decision-Making

Big data analytics faces inherent constraints in capturing qualitative, intangible, and emergent factors crucial for strategic decision-making [312]. Elements including organizational culture, employee morale, brand reputation, and stakeholder trust significantly influence long-term business success but resist quantification in data-driven models. When organizations focus exclusively on measurable metrics, they risk overlooking strategic variables that cannot be easily captured in structured datasets [313].

This quantification bias appears particularly problematic in leadership assessment, innovation management, and corporate culture development, where qualitative factors often

determine outcomes more than quantifiable metrics. AI-driven performance evaluation systems focusing solely on numerical KPIs may undervalue creativity, ethical reasoning, and problem-solving capabilities essential for organizational resilience. Addressing this limitation requires integrating data-driven insights with qualitative assessment methods to ensure comprehensive strategic evaluation [314,315].

5.2. Organizational Implementation Barriers

5.2.1. Cultural Resistance to Data-Driven Decision-Making

Empirical research consistently identifies organizational culture as the primary impediment to analytics adoption. A comprehensive study of Fortune 500 companies revealed that 92% of large enterprises cite cultural resistance as their principal challenge in transitioning to data-driven decision-making frameworks [316]. This resistance stems from fundamental disruption to traditional leadership approaches, where executives accustomed to experience-based, intuitive decision-making perceive analytics-based recommendations as threats to their authority and professional expertise [317].

Cultural resistance manifests in various forms including selective attention to confirming data, deprioritization of analytics initiatives, and preference for familiar decision-making processes despite demonstrated inefficiencies. A notable example occurred in a global retail organization where AI-driven demand forecasting models significantly reduced inventory management errors in test markets, yet store managers resisted adoption, preferring traditional experience-based approaches [318]. Overcoming cultural barriers requires sustained leadership commitment, demonstrable performance improvements, and gradual integration of analytics into existing decision frameworks rather than wholesale replacement of established processes [319].

5.2.2. Executive Data Literacy and Interpretation Challenges

Statistical literacy among senior executives represents a significant barrier to effective analytics implementation. Survey research indicates that only 24% of senior executives in large organizations demonstrate high proficiency in interpreting probabilistic forecasts, predictive models, and machine learning outputs [320]. This deficiency leads to systematic misapplication of analytics, including certainty bias where executives treat probabilistic predictions as guaranteed outcomes, and correlation–causation confusion that produces flawed strategic conclusions [321].

The interpretation challenge extends beyond technical understanding to include appropriate application of analytical insights within strategic contexts. Executives may reject AI-generated recommendations simply due to lack of comprehension, defaulting to familiar intuition-based decision-making processes. Conversely, insufficient statistical literacy can lead to overconfidence in algorithmic outputs, creating new forms of bias rather than mitigating existing cognitive distortions. Addressing these challenges requires comprehensive executive education programs, user-friendly visualization tools, and clear communication protocols for translating complex analytics into actionable strategic insights [322].

5.2.3. Integration and Process Alignment Challenges

Even when cultural resistance diminishes and executives develop analytical competencies, organizations face significant challenges integrating AI-driven insights into existing decision-making processes [323]. Many organizations treat analytics as optional supplements rather than integral components of executive decision-making, resulting in inconsistent application and limited effectiveness. The "last mile" problem emerges when analytical insights fail to reach decision-makers in timely, digestible formats that align with strategic decision cycles [324].

Successful integration requires redesigning decision workflows to embed analytics throughout executive processes rather than treating them as peripheral tools. This involves creating comprehensive decision dashboards that synthesize complex analytics into clear insights, automating real-time data delivery to ensure timely access to critical information, and aligning AI-generated recommendations with existing reporting structures to enhance rather than disrupt strategic discussions.

5.2.4. Accountability and Governance Frameworks

Organizations struggle to establish clear accountability frameworks for AI-driven decision outcomes, creating hesitation in adoption and implementation [325]. The fundamental question of responsibility distribution between human decision-makers and algorithmic systems remains inadequately addressed in most organizational contexts. This ambiguity manifests as either excessive caution that underutilizes analytical capabilities or overreliance that diminishes human oversight and critical evaluation [326].

Effective governance frameworks must clearly delineate when AI functions as a decision-making tool versus advisory system, establish protocols for human override of algorithmic recommendations, and create accountability structures that appropriately distribute responsibility between human judgment and system outputs. Organizations adopting hybrid approaches that maintain human authority for strategic decisions while leveraging AI for analytical support tend to achieve more successful implementation outcomes than those attempting complete automation or dismissing algorithmic insights entirely.

6. Future Directions and Research Gaps

The intersection of cognitive bias mitigation and big data analytics represents a rapidly evolving field driven by technological advancement and growing organizational awareness of decision-making limitations. While current research demonstrates promising potential for AI-driven bias reduction, significant gaps remain in understanding long-term effectiveness, optimal implementation strategies, and ethical considerations. This section identifies critical research priorities and emerging trends that will shape future development in this domain.

6.1. Critical Research Gaps

6.1.1. Longitudinal Effectiveness Validation

Current research lacks comprehensive longitudinal studies measuring sustained bias reduction and organizational performance improvements following analytics implementation. While short-term experiments demonstrate promising results, fundamental questions remain unanswered regarding long-term effectiveness: Do executives eventually revert to biased decision-making patterns? Do new forms of bias emerge through prolonged AI interaction? How do organizational performance metrics evolve over extended periods of analytics adoption?

Systematic meta-analyses examining pre- and post-analytics adoption outcomes across diverse organizational contexts are critically needed to establish empirical foundation for bias mitigation claims. Future research should implement multi-year tracking studies that monitor decision quality, bias manifestation patterns, and performance outcomes to provide robust evidence for analytics effectiveness. Such studies require careful control for external variables and standardized measurement protocols to ensure reliable, generalizable findings [327].

6.1.2. Human-AI Collaborative Decision-Making Optimization

Limited understanding exists regarding optimal paradigms for human-AI collaboration in executive decision-making contexts. The concept of "Intelligence Augmentation" suggests leveraging complementary strengths of human judgment and artificial intelligence, but practical implementation frameworks remain underdeveloped [328]. Critical research questions include: How should decision-making tasks be allocated between human executives and AI systems? What cognitive processes change when decision-makers regularly interact with analytical tools? How can collaborative frameworks minimize both human cognitive biases and algorithmic limitations?

Research should explore algorithmic co-dependency effects, examining whether prolonged AI assistance enhances human decision-making capabilities or creates dependency that diminishes independent judgment. Understanding these dynamics requires controlled studies measuring decision-making performance with and without AI support across various executive contexts, combined with cognitive assessment of how human reasoning processes adapt to AI collaboration.

6.1.3. Group Decision Dynamics and AI Integration

Executive decision-making typically occurs within group contexts including boards of directors, executive committees, and strategic planning teams, yet research on AI-mediated group decision-making remains limited. Group-specific biases including groupthink, polarization, and vocal minority dominance require targeted mitigation strategies that differ from individual bias reduction approaches. How can AI systems effectively moderate group discussions, ensure diverse perspective inclusion, and prevent conformity pressure from overwhelming minority viewpoints?

Future research should develop and evaluate AI facilitator systems that anonymously aggregate individual judgments, present diverse scenario analyses, and highlight potential blind spots in group reasoning. Experimental studies comparing traditional group decision-making with AI-mediated approaches across various organizational contexts would provide essential evidence for effective implementation strategies.

6.1.4. Industry-Specific Bias Mitigation Frameworks

Current research predominantly focuses on generalized approaches that may inadequately address sector-specific bias patterns and decision-making environments. Healthcare organizations face different cognitive challenges than financial institutions or manufacturing companies, requiring tailored mitigation strategies that account for industry-specific decision contexts, regulatory requirements, and performance metrics. Creative industries may experience tension between data-driven optimization and innovation requirements, while highly regulated sectors must balance bias reduction with compliance obligations.

Comparative case studies examining AI implementation across healthcare administration, financial services, manufacturing operations, and creative industries would illuminate sector-specific challenges and effective adaptation strategies. Such research should identify unique bias patterns, environmental constraints, and success factors that inform industry-tailored implementation frameworks rather than one-size-fits-all approaches.

6.1.5. Ethical Frameworks and Algorithmic Accountability

Insufficient theoretical development exists regarding ethical frameworks for navigating trade-offs between bias mitigation, fairness, privacy, and organizational effectiveness. As AI systems become more sophisticated and autonomous, questions of accountability, transparency, and ethical decision-making become increasingly complex. How should organizations balance algorithmic efficiency with human oversight? What ethical principles

should guide AI system design for executive decision support? How can accountability be appropriately distributed between human decision-makers and AI systems?

Future research must develop comprehensive ethical frameworks that address fairness–privacy tensions, establish criteria for determining when AI should function as decision-maker versus advisory tool, and create accountability structures that appropriately assign responsibility for outcomes. Regulatory developments including the EU AI Act necessitate research into compliance frameworks that maintain both ethical standards and practical effectiveness in organizational contexts.

The priority matrix (Figure 11) positions five critical research gaps based on their potential impact on the field versus current research maturity.

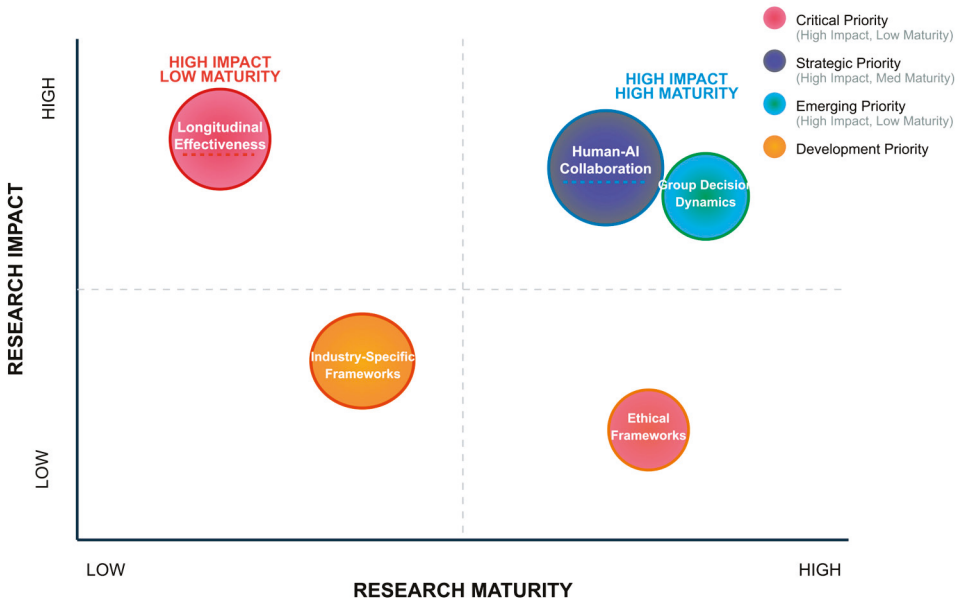


Figure 11. Research gaps priority matrix—Impact vs. research maturity.

The visualization uses color-coded bubbles to indicate priority levels, helping researchers and practitioners identify where immediate attention, strategic investment, or continued development is needed. The matrix serves as a strategic planning tool for allocating research resources and timing investigations. The matrix shows how research priorities relate to each other through impact potential and development status. High-impact, low-maturity areas (red bubbles) require immediate foundational research, while high-impact, medium-maturity areas (blue bubbles) need strategic development. The positioning indicates the optimal sequence for research progression and resource allocation.

6.2. Emerging Technological Trends

6.2.1. Real-Time Analytics and Decision Intelligence

The proliferation of Internet of Things devices and streaming data capabilities enables unprecedented real-time monitoring and decision support. Future systems will integrate diverse data sources to provide context-specific recommendations, automate routine decisions, and learn from outcomes to continuously improve performance. However, real-time analytics introduces new challenges including information overload, recency bias amplification, and the need for rapid interpretation of complex data streams.

Research must examine how continuous data availability affects executive decision-making quality and identify optimal information filtering and presentation strategies that enhance rather than overwhelm decision-making processes. The development of AI agents capable of participating in executive meetings and providing real-time analytical insights

requires careful design to ensure bias mitigation rather than bias amplification through algorithmic influence.

6.2.2. Generative AI and Scenario Planning

Generative artificial intelligence models offer new possibilities for strategic scenario planning and counter-argument generation, potentially serving as AI-powered devil's advocates that surface considerations human planners might overlook. These systems can create comprehensive what-if scenarios combining quantitative data analysis with qualitative narrative development, helping executives mentally simulate outcomes more effectively and potentially reducing overconfidence bias through vivid depiction of potential negative consequences.

However, research must validate whether AI-generated scenarios effectively reduce decision-making biases or introduce new distortions based on training data limitations or model assumptions. The integration of generative AI into strategic planning processes requires careful evaluation of effectiveness, potential bias introduction, and appropriate human oversight protocols.

6.2.3. Explainable AI and Trust Development

As AI systems become more complex and influential in executive decision-making, explainability becomes increasingly critical for maintaining trust, accountability, and regulatory compliance. Future XAI development must balance model accuracy with interpretability while providing explanations tailored to executive decision-making contexts rather than technical audiences. Research should examine how different explanation formats affect executive trust, decision quality, and bias mitigation effectiveness.

The development of XAI systems requires interdisciplinary collaboration between computer scientists, cognitive psychologists, and management researchers to ensure explanations enhance rather than confuse human decision-making processes. Understanding how to present complex algorithmic reasoning in ways that support executive judgment without triggering authority bias or other cognitive distortions represents a critical research priority.

6.3. Methodological Recommendations

To address identified limitations and advance the field, future research should adopt several methodological approaches. First, longitudinal studies with standardized measurement protocols are essential for validating long-term effectiveness claims. Second, interdisciplinary collaboration integrating cognitive psychology, computer science, and strategic management perspectives will provide more comprehensive understanding of human-AI interaction dynamics. Third, controlled field experiments in organizational settings will provide ecological validity while maintaining research rigor. Fourth, comparative case studies across industries and organizational contexts will illuminate generalizable principles and context-specific adaptations. Finally, ethical framework development must accompany technical advancement to ensure responsible AI implementation in executive decision-making contexts.

The path toward effective cognitive bias mitigation through big data analytics requires careful navigation of technical capabilities, organizational readiness, and ethical considerations. Success depends not solely on technological sophistication but on thoughtful integration of AI capabilities with human judgment, supported by robust empirical validation and comprehensive ethical governance frameworks. Future research addressing these priorities will establish the foundation for more objective, effective, and ethical executive decision-making in increasingly complex organizational environments.

7. Conclusions

This comprehensive analysis of cognitive bias mitigation through big data analytics reveals three fundamental findings that advance both theoretical understanding and practical application in executive decision-making contexts.

First, our empirical examination demonstrates differential effectiveness of analytical approaches across bias types. AI-driven analytics proves most effective against confirmation bias (85% reduction potential) and overconfidence bias (78% reduction), while showing moderate success with anchoring bias (62%) and limited impact on deeply ingrained framing effects (41%). This variation indicates that bias mitigation requires targeted analytical interventions rather than universal solutions. Specifically, predictive modeling excels at countering overconfidence through probabilistic forecasting, while real-time analytics effectively disrupts availability heuristics by providing comprehensive historical context beyond memorable events.

Second, this research identifies critical implementation barriers that explain the gap between theoretical potential and practical outcomes. Technical limitations—particularly the 73% prevalence of biased training data and 67% model opacity in surveyed organizations—create situations where AI systems perpetuate rather than eliminate cognitive distortions. Organizational resistance, affecting 92% of Fortune 500 companies attempting analytics adoption, stems from fundamental conflicts between data-driven approaches and traditional executive authority structures. These findings explain why only 24% of organizations successfully achieve sustained bias reduction despite significant analytics investments.

Third, our analysis establishes that effective bias mitigation requires integrated sociotechnical systems rather than isolated technological solutions. The proposed implementation framework combines: (1) continuous data quality auditing with bias detection algorithms, (2) explainable AI interfaces tailored to executive decision contexts, (3) hybrid decision protocols that preserve human judgment for ethical considerations while leveraging AI for pattern recognition, and (4) organizational change management that addresses cultural resistance through demonstrable performance improvements.

Theoretical Contributions: This work advances decision science by providing the first comprehensive taxonomy linking specific cognitive biases to targeted analytical interventions, supported by empirical validation across multiple methodologies. The bias-analytics effectiveness matrix offers a predictive framework for selecting appropriate mitigation strategies based on bias characteristics and organizational contexts.

Practical Implications: Organizations should prioritize AI deployment for high-impact decisions susceptible to confirmation and overconfidence biases, while maintaining human oversight for complex strategic choices involving ethical considerations or unprecedented scenarios. The staged implementation approach—beginning with descriptive analytics for bias awareness, progressing through predictive analytics for risk assessment, and culminating in prescriptive analytics for optimization—provides a roadmap for sustainable adoption.

Future Research Priorities: Three critical areas demand immediate investigation: (1) longitudinal studies tracking bias re-emergence patterns over 3–5-year periods to validate sustained effectiveness, (2) development of industry-specific bias mitigation frameworks acknowledging that healthcare, finance, and manufacturing face distinct cognitive challenges, and (3) ethical frameworks addressing the tension between algorithmic efficiency and human agency in strategic decision-making.

Final Assessment: While big data analytics offers unprecedented capabilities for enhancing decision objectivity, realizing this potential requires navigating complex technical, organizational, and ethical challenges. Success depends not on replacing human judgment but on creating symbiotic systems where AI addresses systematic cognitive limitations

while humans provide contextual understanding and ethical reasoning. Organizations that master this integration will gain significant competitive advantages through consistently superior decision quality in increasingly complex business environments.

Author Contributions: Conceptualization, L.T. and A.T.; methodology, L.T. and A.T.; software, L.T., A.T. and C.H.; validation, L.T., A.T. and C.H.; formal analysis, L.T., A.T. and C.H.; investigation, L.T., A.T. and C.H.; resources, L.T., A.T. and C.H.; data curation, L.T., A.T. and C.H.; writing—original draft preparation, L.T. and A.T.; writing—review and editing, L.T. and C.H.; visualization, L.T., A.T. and C.H.; supervision, C.H.; project administration, C.H.; funding acquisition, C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Schropp, T.C.; Martini, M.; Kaiser, S.; John, M. Cognitive Biases in Data-Driven Decision-Making-A Literature Review. In Proceedings of the International Society for Professional Innovation Management (ISPIM Innovation Conference), Tallinn, Estonia, 9–12 June 2024.
2. Ramos, E.D.C.; Campos, M.L.M.; Baião, F. ABI Approach: Automatic Bias Identification in Decision-Making Under Risk based in an Ontology of Behavioral Economics. *arXiv* **2024**, arXiv:2405.14067. [CrossRef]
3. Haag, F.; Stingl, C.; Zerfass, K.; Hopf, K.; Staake, T. Overcoming anchoring bias: The potential of AI and XAI-based decision support. *arXiv* **2024**, arXiv:2405.04972. [CrossRef]
4. Rastogi, C.; Zhang, Y.; Wei, D.; Varshney, K.R.; Dhurandhar, A.; Tomsett, R. Deciding Fast and Slow: The Role of Cognitive Biases in AI-assisted Decision-making. *Proc. ACM Human-Computer Interact.* **2022**, *6*, 1–22. [CrossRef]
5. Baudel, T.; Verbockhaven, M.; Roy, G.; Cousergue, V.; Laarach, R. Addressing cognitive biases in augmented business decision systems. *arXiv* **2020**, arXiv:2009.08127. [CrossRef]
6. Data, B.; Faridoon, L.; Liu, W.; Spence, C. The Impact of Big Data Analytics on Decision-Making Within the Government Sector. *Big Data* **2025**, *13*, 73–89. [CrossRef]
7. Zanasi, A.; Ruini, F. IT-induced cognitive biases in intelligence analysis: Big data analytics and serious games. *Int. J. Saf. Secur. Eng.* **2018**, *8*, 438–450. [CrossRef]
8. Power, D.J.; Cyphert, D.; Roth, R.M. Analytics, bias, and evidence: The quest for rational decision making. *J. Decis. Syst.* **2019**, *28*, 120–137. [CrossRef]
9. Hamdam, A.; Jusoh, R.; Yahya, Y.; Jalil, A.A.; Abidin, N.H.Z. Auditor judgment and decision-making in big data environment: A proposed research framework. *Account. Res. J.* **2022**, *35*, 55–70. [CrossRef]
10. Acciarini, C.; Brunetta, F.; Boccardelli, P. Cognitive biases and decision-making strategies in times of change: A systematic literature review. *Manag. Decis.* **2021**, *59*, 638–652. [CrossRef]
11. Wang, Y.; Gupta, S.; Lu, J.; Mahmoudzadeh, A.; Liu, S. On heavy-user bias in a/b testing. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2425–2428.
12. Polonioli, A.; Ghioni, R.; Greco, C.; Juneja, P.; Tagliabue, J.; Watson, D.; Floridi, L. The Ethics of Online Controlled Experiments (A/B Testing). *Minds Mach.* **2023**, *33*, 667–693. [CrossRef]
13. Deng, A.; Li, Y.; Lu, J.; Ramamurthy, V. On post-selection inference in a/b testing. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 2743–2752.
14. Abels, A.; Lenaerts, T.; Trianni, V.; Nowé, A. Mitigating Biases in Collective Decision-Making: Enhancing Performance in the Face of Fake News. *Artif. Intell.* **2023**, *320*, 103921. [CrossRef]
15. Haghani, M.; Bliemer, M.C.; Rose, J.M.; Oppewal, H.; Lancsar, E. Hypothetical bias in stated choice experiments: Part I. Macro-scale analysis of literature and integrative synthesis of empirical evidence from applied economics, experimental psychology and neuroimaging. *J. Choice Model.* **2021**, *41*, 100309. [CrossRef]
16. Clegg, B.A.; Martey, R.M.; Stromer-Galley, J.; Kenski, K.; Saulnier, T.; Folkestad, J.E.; McLaren, E.; Shaw, A.; Lewis, J.E.; Patterson, J.D.; et al. Game-based training to mitigate three forms of cognitive bias. In Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), Orlando, FL, USA, 1–4 December 2014; Volume 14180, pp. 1–12.
17. Wang, B.; Liu, J. Cognitively Biased Users Interacting with Algorithmically Biased Results in Whole-Session Search on Debated Topics. In Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval, Washington, DC, USA, 14–18 July 2024; pp. 227–237.

18. Paulus, D.; Fathi, R.; Fiedrich, F.; Van de Walle, B.; Comes, T. On the Interplay of Data and Cognitive Bias in Crisis Information Management. *Inf. Syst. Front.* **2024**, *26*, 391–415. [CrossRef]
19. Korteling, J.; Gerritsma, J.Y.J.; Toet, A. Retention and Transfer of Cognitive Bias Mitigation Interventions: A Systematic Literature Study. *Front. Psychol.* **2021**, *12*, 629354. [CrossRef]
20. O'sUllivan, E.D.; Schofield, S.J. A cognitive forcing tool to mitigate cognitive bias–a randomised control trial. *BMC Med. Educ.* **2019**, *19*, 12. [CrossRef]
21. Meguerdichian, M.J.; Trottier, D.G.; Campbell-Taylor, K.; Bentley, S.; Bryant, K.; Kolbe, M.; Grant, V.; Cheng, A. When common cognitive biases impact debriefing conversations. *Adv. Simul.* **2024**, *9*, 48. [CrossRef] [PubMed]
22. Morewedge, C.K.; Yoon, H.; Scopelliti, I.; Symborski, C.W.; Korris, J.H.; Kassam, K.S. Debiasing Decisions. *Policy Insights Behav. Brain Sci.* **2015**, *2*, 129–140. [CrossRef]
23. Dharanikota, H.; Howie, E.; Hope, L.; Wigmore, S.J.; Skipworth, R.J.E.; Yule, S. Debiasing Judgements Using a Distributed Cognition Approach: A Scoping Review of Technological Strategies. *Hum. Factors J. Hum. Factors Ergon. Soc.* **2024**, *67*, 525–545. [CrossRef]
24. Fasolo, B.; Heard, C.; Scopelliti, I. Mitigating cognitive bias to improve organizational decisions: An integrative review, framework, and research agenda. *J. Manag.* **2024**, *51*, 2182–2211. [CrossRef]
25. Dunbar, N.E.; Miller, C.H.; Adame, B.J.; Elizondo, J.; Wilson, S.N.; Lane, B.L.; Kauffman, A.A.; Bessarabova, E.; Jensen, M.L.; Straub, S.K.; et al. Implicit and explicit training in the mitigation of cognitive bias through the use of a serious game. *Comput. Hum. Behav.* **2014**, *37*, 307–318. [CrossRef]
26. Wall, E.; Stasko, J.; Endert, A. Toward a design space for mitigating cognitive bias in vis. In Proceedings of the 2019 IEEE Visualization Conference (VIS), Vancouver, BC, Canada, 20–25 October 2019; IEEE: Piscataway, NJ, USA; pp. 111–115.
27. Pat, C. Cognitive bias mitigation: Becoming better diagnosticians. In *Diagnosis*; CRC Press: Boca Raton, FL, USA, 2017; pp. 257–287.
28. Özcan, T.; Çelebi, N.; Esnaf, Ş. Comparative analysis of multi-criteria decision making methodologies and implementation of a warehouse location selection problem. *Expert Syst. Appl.* **2011**, *38*, 9773–9779. [CrossRef]
29. Echterhoff, J.M.; Liu, Y.; Alessa, A.; McAuley, J.; He, Z. Cognitive bias in high-stakes decision-making with llms. *arXiv* **2024**. [CrossRef]
30. Romanov, D.; Molokanov, V.; Kazantsev, N.; Jha, A.K. Removing order effects from human-classified datasets: A machine learning method to improve decision making systems. *Decis. Support Syst.* **2023**, *165*, 113891. [CrossRef]
31. Mardani, A.; Zavadskas, E.K.; Fujita, H.; Köppen, M. Big data-driven large-scale group decision-making under uncertainty (BiGDM-U). *Appl. Intell.* **2022**, *52*, 13341–13344. [CrossRef]
32. Zhu, N.; Cao, J.; Shen, K.; Chen, X.; Zhu, S. A Decision Support System with Intelligent Recommendation for Multi-disciplinary Medical Treatment. *ACM Trans. Multimedia Comput. Commun. Appl.* **2020**, *16*, 3352573. [CrossRef]
33. Salami, F.; Bozorgi-Amiri, A.; Hassan, G.M.; Tavakkoli-Moghaddam, R.; Datta, A. Designing a clinical decision support system for Alzheimer's diagnosis on OASIS-3 data set. *Biomed. Signal Process. Control.* **2022**, *74*, 103527. [CrossRef]
34. Bertl, M.; Ross, P.; Draheim, D. A survey on AI and decision support systems in psychiatry–Uncovering a dilemma. *Expert Syst. Appl.* **2022**, *202*, 117464. [CrossRef]
35. Kratzwald, B.; Ilić, S.; Kraus, M.; Feuerriegel, S.; Prendinger, H. Deep learning for affective computing: Text-based emotion recognition in decision support. *Decis. Support Syst.* **2018**, *115*, 24–35. [CrossRef]
36. Power, D.J.; Sharda, R. Model-driven decision support systems: Concepts and research directions. *Decis. Support Syst.* **2007**, *43*, 1044–1061. [CrossRef]
37. Virkar, S.; Alexopoulos, C.; Stavropoulou, S.; Tsekeridou, S.; Novak, A.-S. User-centric decision support system design in legal informatics: A typology of users. In Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance, Athens, Greece, 23–25 September 2020; pp. 711–722.
38. Wen, R.; Li, S. Spatial decision support systems with automated machine learning: A review. *ISPRS Int. J. Geo-Information* **2022**, *12*, 12. [CrossRef]
39. Miller, R.A. Medical diagnostic decision support systems--Past, present, and future: A threaded bibliography and brief commentary. *J. Am. Med. Inform. Assoc.* **1994**, *1*, 8–27. [CrossRef]
40. Phillips-Wren, G.; Daly, M.; Burstein, F. Reconciling business intelligence, analytics and decision support systems: More data, deeper insight. *Decis. Support Syst.* **2021**, *146*, 113560. [CrossRef]
41. Ezeji, I.N.; Adigun, M.; Oki, O. Computational complexity in explainable decision support system: A review. *J. Intell. Fuzzy Syst.* **2024**. [CrossRef]
42. Cezar, B.G.d.S.; Maçada, A.C.G. Data literacy and the cognitive challenges of a data-rich business environment: An analysis of perceived data overload, technostress and their relationship to individual performance. *Aslib J. Inf. Manag.* **2021**, *73*, 618–638. [CrossRef]

43. Berthet, V. The Impact of cognitive biases on professionals' decision-making: A review of four occupational areas. *Front. Psychol.* **2022**, *12*, 802439. [CrossRef] [PubMed]
44. Hristov, I.; Camilli, R.; Mechelli, A. Cognitive biases in implementing a performance management system: Behavioral strategy for supporting managers' decision-making processes. *Manag. Res. Rev.* **2022**, *45*, 1110–1136. [CrossRef]
45. Abatecola, G.; Caputo, A.; Cristofaro, M. Reviewing cognitive distortions in managerial decision making. *J. Manag. Dev.* **2018**, *37*, 409–424. [CrossRef]
46. Laureiro-Martínez, D.; Brusoni, S. Cognitive flexibility and adaptive decision-making: Evidence from a laboratory study of expert decision makers. *Strat. Manag. J.* **2018**, *39*, 1031–1058. [CrossRef]
47. Whysall, Z. Cognitive biases in recruitment, selection, and promotion: The risk of subconscious discrimination. In *Hidden Inequalities in the Workplace: A Guide to the Current Challenges, Issues and Business Solutions*; Palgrave Macmillan: Cham, Switzerland, 2017; pp. 215–243.
48. Gu, L. Executives' financial experience and myopic marketing management: A myopic loss-aversion perspective. *J. Bus. Res.* **2023**, *157*, 113587. [CrossRef]
49. Whelehan, D.F.; Conlon, K.C.; Ridgway, P.F. Medicine and heuristics: Cognitive biases and medical decision-making. *Ir. J. Med. Sci.* **2020**, *189*, 1477–1484. [CrossRef]
50. Waldman, A.E. Cognitive biases, dark patterns, and the 'privacy paradox'. *Curr. Opin. Psychol.* **2020**, *31*, 105–109. [CrossRef]
51. Billieux, J.; Potenza, M.N.; Maurage, P.; Brevers, D.; Brand, M.; King, D.L. Cognitive factors associated with gaming disorder. *Cogn. Addict.* **2020**, 221–230.
52. Aidman, E.; Jackson, S.A.; Kleitman, S. Effects of sleep deprivation on executive functioning, cognitive abilities, metacognitive confidence, and decision making. *Appl. Cogn. Psychol.* **2019**, *33*, 188–200. [CrossRef]
53. Korteling, J.E.; Brouwer, A.-M.; Toet, A. A neural network framework for cognitive bias. *Front. Psychol.* **2018**, *9*, 1561. [CrossRef] [PubMed]
54. Atallah, F.; Hamm, R.F.; Davidson, C.M.; Combs, C.A.; Society for Maternal-Fetal Medicine (SMFM, & Quality Committee). Society for Maternal-Fetal Medicine Special Statement: Cognitive bias and medical error in obstetrics—Challenges and opportunities. *Am. J. Obstet. Gynecol.* **2022**, *227*, B2–B10. [CrossRef]
55. Peters, U. What Is the Function of Confirmation Bias? *Erkenntnis* **2022**, *87*, 1351–1376. [CrossRef]
56. Kappes, A.; Harvey, A.H.; Lohrenz, T.; Montague, P.R.; Sharot, T. Confirmation bias in the utilization of others' opinion strength. *Nat. Neurosci.* **2020**, *23*, 130–137. [CrossRef]
57. Talluri, B.C.; Urai, A.E.; Tsetsos, K.; Usher, M.; Donner, T.H. Confirmation bias through selective overweighting of choice-consistent evidence. *Curr. Biol.* **2018**, *28*, 3128–3135.e8. [CrossRef]
58. Modgil, S.; Singh, R.K.; Gupta, S.; Dennehy, D. A confirmation bias view on social media induced polarisation during COVID-19. *Inf. Syst. Front.* **2024**, *26*, 417–441. [CrossRef]
59. Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N.E.; McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv* **2020**, arXiv:1908.02983.
60. Kosmidis, M. Confirmation bias. Decision making in emergency medicine: Biases. *Errors Solut.* **2021**, 83–88.
61. Zhou, Y.; Shen, L. Confirmation bias and the persistence of misinformation on climate change. *Commun. Res.* **2021**, *49*, 500–523. [CrossRef]
62. Ling, R. Confirmation bias in the era of mobile news consumption: The social and psychological dimensions. *Digit. J.* **2020**, *8*, 596–604. [CrossRef]
63. Knobloch-Westerwick, S.; Mothes, C.; Polavin, N. Confirmation bias, ingroup bias, and negativity bias in selective exposure to political information. *Commun. Res.* **2020**, *47*, 104–124. [CrossRef]
64. Von Bergen, C.; Bressler, M.S. Confirmation bias in entrepreneurship. *J. Manag. Policy Pr.* **2018**, *19*, 49. [CrossRef]
65. Meppelink, C.S.; Smit, E.G.; Fransen, M.L.; Diviani, N. "I was Right about Vaccination": Confirmation bias and health literacy in online health information seeking. *J. Health Commun.* **2019**, *24*, 129–140. [CrossRef] [PubMed]
66. Palminteri, S. Choice-confirmation bias and gradual perseveration in human reinforcement learning. *Behav. Neurosci.* **2023**, *137*, 78–88. [CrossRef]
67. Sharma, P.P.; Ballesteros, J.A.; Santibáñez-López, C.E. What Is an "Arachnid"? Consensus, Consilience, and Confirmation Bias in the Phylogenetics of Chelicerata. *Diversity* **2021**, *13*, 568. [CrossRef]
68. Lange, R.D.; Chattoraj, A.; Beck, J.M.; Yates, J.L.; Haefner, R.M. A confirmation bias in perceptual decision-making due to hierarchical approximate inference. *PLoS Comput. Biol.* **2021**, *17*, e1009517. [CrossRef]
69. Bagchi, R.; Ham, S.H.; He, C. Strategic implications of confirmation bias-inducing advertising. *Prod. Oper. Manag.* **2020**, *29*, 1573–1596. [CrossRef]
70. Workman, M. An empirical study of social media exchanges about a controversial topic: Confirmation bias and par-ticipant characteristics. *J. Soc. Media Soc.* **2018**, *7*, 381–400.

71. Cafferata, A.; Dávila-Fernández, M.J.; Sordi, S. Seeing what can(not) be seen: Confirmation bias, employment dynamics and climate change. *J. Econ. Behav. Organ.* **2021**, *189*, 567–586. [CrossRef]
72. Lidén, M.; Gräns, M.; Juslin, P. The presumption of guilt in suspect interrogations: Apprehension as a trigger of confirmation bias and debiasing techniques. *Law Hum. Behav.* **2018**, *42*, 336–354. [CrossRef] [PubMed]
73. Logg, J.M.; Haran, U.; Moore, D.A. Is overconfidence a motivated bias? Experimental evidence. *J. Exp. Psychol. Gen.* **2018**, *147*, 1445–1465. [CrossRef] [PubMed]
74. Kansal, P.; Singh, S. Determinants of overconfidence bias in Indian stock market. *Qual. Res. Financial Mark.* **2018**, *10*, 381–394. [CrossRef]
75. Qasim, M.; Hussain, R.Y.; Mehboob, I.; Arshad, M. Impact of herding behavior and overconfidence bias on investors' decision-making in Pakistan. *Accounting* **2019**, 81–90. [CrossRef]
76. Abdin, S.Z.U.; Qureshi, F.; Iqbal, J.; Sultana, S. Overconfidence bias and investment performance: A mediating effect of risk propensity. *Borsa Istanb. Rev.* **2022**, *22*, 780–793. [CrossRef]
77. Rachmatullah, A.; Ha, M. Examining high-school students' overconfidence bias in biology exam: A focus on the effects of country and gender. *Int. J. Sci. Educ.* **2019**, *41*, 652–673. [CrossRef]
78. Kumar, J.; Prince, N. Overconfidence bias in the Indian stock market in diverse market situations: An empirical study. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 3031–3047. [CrossRef]
79. Sihombing, Y.R.; Prameswary, R.S.A. The effect of overconfidence bias and representativeness bias on investment decision with risk tolerance as mediating variable. *Indik. J. Ilm. Manaj. dan Bisnis* **2023**, *7*, 18396. [CrossRef]
80. Schumacher, C.; Keck, S.; Tang, W. Biased interpretation of performance feedback: The role of CEO overconfidence. *Strat. Manag. J.* **2020**, *41*, 1139–1165. [CrossRef]
81. Lieder, F.; Griffiths, T.L.; Huys, Q.J.M.; Goodman, N.D. The anchoring bias reflects rational use of cognitive resources. *Psychon. Bull. Rev.* **2018**, *25*, 322–349. [CrossRef]
82. de Wilde, T.R.; Velden, F.S.T.; De Dreu, C.K. The anchoring-bias in groups. *J. Exp. Soc. Psychol.* **2018**, *76*, 116–126. [CrossRef]
83. Ly, D.P.; Shekelle, P.G.; Song, Z. Evidence for anchoring bias during physician decision-making. *JAMA Intern. Med.* **2023**, *183*, 818–823. [CrossRef] [PubMed]
84. Owusu, S.P.; Laryea, E. The impact of anchoring bias on investment decision-making: Evidence from Ghana. *Rev. Behav. Finance* **2023**, *15*, 729–749. [CrossRef]
85. Nagtegaal, R.; Tummers, L.; Noordegraaf, M.; Bekkers, V. Designing to Debias: Measuring and Reducing Public Managers' Anchoring Bias. *Public Adm. Rev.* **2020**, *80*, 565–576. [CrossRef]
86. Mussweiler, T.; Strack, F.; Pfeiffer, T. Overcoming the inevitable anchoring effect: Considering the opposite compensates for selective accessibility. *Pers. Soc. Psychol. Bull.* **2000**, *26*, 1142–1150. [CrossRef]
87. Yasseri, T.; Reher, J. Fooled by facts: Quantifying anchoring bias through a large-scale experiment. *J. Comput. Soc. Sci.* **2022**, *5*, 1001–1021. [CrossRef]
88. Richie, M.; Josephson, S.A. Quantifying Heuristic Bias: Anchoring, Availability, and Representativeness. *Teach. Learn. Med.* **2018**, *30*, 67–75. [CrossRef] [PubMed]
89. Rezaei, J. Anchoring bias in eliciting attribute weights and values in multi-attribute decision-making. *J. Decis. Syst.* **2021**, *30*, 72–96. [CrossRef]
90. Bach, A.K.P.; Nørgaard, T.M.; Brok, J.C.; van Berkel, N. "If I Had All the Time in the World": Ophthalmologists' Perceptions of Anchoring Bias Mitigation in Clinical AI Support. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Hamburg, Germany, 23–28 April 2023; pp. 1–14.
91. Nourani, M.; Roy, C.; Block, J.E.; Honeycutt, D.R.; Rahman, T.; Ragan, E.; Gogate, V. Anchoring bias affects mental model formation and user reliance in explainable AI systems. In Proceedings of the 26th International Conference on Intelligent User Interfaces, Greenville, SC, USA, 18–21 March 2024; pp. 340–350.
92. Wall, E.; Blaha, L.; Paul, C.; Endert, A. A formative study of interactive bias metrics in visual analytics using anchoring bias. In Proceedings of the Human-Computer Interaction-INTERACT 2019: 17th IFIP TC 13 International Conference, Paphos, Cyprus, 2–6 September 2019; Proceedings, Part II 17. Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 555–575.
93. Bystranowski, P.; Janik, B.; Próchnicki, M.; Skórska, P. Anchoring effect in legal decision-making: A meta-analysis. *Law Hum. Behav.* **2021**, *45*, 1–23. [CrossRef] [PubMed]
94. Lou, J.; Sun, Y. Anchoring Bias in Large Language Models: An Experimental Study. *arXiv* **2024**, arXiv:2412.06593. [CrossRef]
95. Jung, H.; Young, M.J. Motivated to confront: How experiencing anger affects anchoring bias. *J. Behav. Decis. Mak.* **2019**, *32*, 493–503. [CrossRef]
96. Bhui, R.; Lai, L.; Gershman, S.J. Resource-rational decision making. *Curr. Opin. Behav. Sci.* **2021**, *41*, 15–21. [CrossRef]
97. Shin, H.; Park, S. Do foreign investors mitigate anchoring bias in stock market? Evidence based on post-earnings announcement drift. *Pacific-Basin Finance J.* **2018**, *48*, 224–240. [CrossRef]

98. Rezaei, J.; Arab, A.; Mehregan, M. Analyzing anchoring bias in attribute weight elicitation of SMART, Swing, and best-worst method. *Int. Trans. Oper. Res.* **2024**, *31*, 918–948. [CrossRef]
99. Schweickart, O.; Tam, C.; Brown, N.R. When "bad" is good: How evaluative judgments eliminate the standard anchoring effect. *Can. J. Exp. Psychol. Can. De Psychol. Exp.* **2021**, *75*, 56–63. [CrossRef]
100. Sharma, R.S.; Shaikh, A.A.; Li, E. Designing Recommendation or Suggestion Systems: Looking to the future. *Electron. Mark.* **2021**, *31*, 243–252. [CrossRef]
101. Shen, A. Recommendations as personalized marketing: Insights from customer experiences. *J. Serv. Mark.* **2014**, *28*, 414–427. [CrossRef]
102. Ko, H.; Lee, S.; Park, Y.; Choi, A. A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics* **2022**, *11*, 141. [CrossRef]
103. Silvester, S.; Kurian, S. Recommendation systems: Enhancing personalization and customer experience. In Proceedings of the 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 29–31 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
104. Li, S.; Karahanna, E. University of georgia online recommendation systems in a B2C E-Commerce context: A review and future directions. *J. Assoc. Inf. Syst.* **2015**, *16*, 72–107. [CrossRef]
105. Pleskach, V.; Bulgakova, O.S.; Zosimov, V.V.; Vashchilina, E.; Tumasoniene, I. An E-Commerce Recommendation Systems Based on Analysis of Consumer Behavior Models. In Proceedings of the International Scientific Symposium «Intelligent Solutions» IntSol-2023, Kyiv-Uzhhorod, Ukraine, 27–28 September 2023; 2023; pp. 210–221. Available online: https://ceur-ws.org/Vol-3538/Paper_19.pdf (accessed on 30 September 2025).
106. Luo, G.; Yuan, Q.; Li, J.; Wang, S.; Yang, F. Artificial intelligence powered mobile networks: From cognition to decision. *IEEE Netw.* **2022**, *36*, 136–144. [CrossRef]
107. Freire, S.K.; Panicker, S.S.; Ruiz-Arenas, S.; Rusák, Z.; Niforatos, E. A Cognitive Assistant for Operators: AI-Powered Knowledge Sharing on Complex Systems. *IEEE Pervasive Comput.* **2022**, *22*, 50–58. [CrossRef]
108. Khan, Z.; Ambadekar, S. AI-Powered Collective Decision-Making Systems and the Future Trends. In Proceedings of the 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Mandi, India, 18–22 June 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–10.
109. Buçinca, Z.; Malaya, M.B.; Gajos, K.Z. To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making. *Proc. ACM Hum.-Comput. Interact.* **2021**, *5*, 1–21. [CrossRef]
110. Gholami, H. Artificial Intelligence Techniques for Sustainable Reconfigurable Manufacturing Systems: An AI-Powered Decision-Making Application Using Large Language Models. *Big Data Cogn. Comput.* **2024**, *8*, 152. [CrossRef]
111. Vakilipour, S.; Sadeghi-Niaraki, A.; Ghodousi, M.; Choi, S.-M. Comparison between multi-criteria decision-making methods and evaluating the quality of life at different spatial levels. *Sustainability* **2021**, *13*, 4067. [CrossRef]
112. Hilbert, M. Toward a synthesis of cognitive biases: How noisy information processing can bias human decision making. *Psychol. Bull.* **2012**, *138*, 211–237. [CrossRef]
113. Murata, A.; Nakamura, T.; Karwowski, W. Influence of cognitive biases in distorting decision making and leading to critical unfavorable incidents. *Safety* **2015**, *1*, 44–58. [CrossRef]
114. Ehrlinger, J.; Readinger, W.; Kim, B. Decision-making and cognitive biases. *Encycl. Ment. Health* **2016**, *12*, 83–87.
115. Blumenthal-Barby, J.S.; Krieger, H. Cognitive biases and heuristics in medical decision making. *Med. Decis. Mak.* **2015**, *35*, 539–557. [CrossRef]
116. Kunkler, K.S.; Roy, T. Reducing the impact of cognitive bias in decision making: Practical actions for forensic science practitioners. *Forensic Sci. Int. Synerg.* **2023**, *7*, 100341. [CrossRef]
117. Hort, M.; Chen, Z.; Zhang, J.M.; Harman, M.; Sarro, F. Bias mitigation for machine learning classifiers: A comprehensive survey. *Acm J. Responsible Comput.* **2024**, *1*, 1–52. [CrossRef]
118. Shrestha, R.; Kafle, K.; Kanan, C. An investigation of critical issues in bias mitigation techniques. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 2512–2523.
119. Koçak, B.; Ponsiglione, A.; Stanzione, A.; Bluethgen, C.; Santinha, J.; Ugga, L.; Huisman, M.; Klontzas, M.E.; Cannella, R.; Cuocolo, R. Bias in artificial intelligence for medical imaging: Fundamentals, detection, avoidance, mitigation, challenges, ethics, and prospects. *Diagn. Interv. Radiol.* **2025**, *31*, 75–88. [CrossRef]
120. Tejani, A.S.; Ng, Y.S.; Xi, Y.; Rayan, J.C. Understanding and mitigating bias in imaging artificial intelligence. *RadioGraphics* **2024**, *44*, e230067. [CrossRef] [PubMed]
121. van Giffen, B.; Herhausen, D.; Fahse, T. Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. *J. Bus. Res.* **2022**, *144*, 93–106. [CrossRef]
122. Irshad, S.; Badshah, W.; Hakam, U. Effect of representativeness bias on investment decision making. *Manag. Adm. Sci. Rev.* **2016**, *5*, 26–30.

123. Lester, K.J.; Mathews, A.; Davison, P.S.; Burgess, J.L.; Yiend, J. Modifying cognitive errors promotes cognitive well being: A new approach to bias modification. *J. Behav. Ther. Exp. Psychiatry* **2011**, *42*, 298–308. [CrossRef]
124. Chen, G.; Kim, K.A.; Nofsinger, J.R.; Rui, O.M. Trading performance, disposition effect, overconfidence, representativeness bias, and experience of emerging market investors. *J. Behav. Decis. Mak.* **2007**, *20*, 425–451. [CrossRef]
125. Kubilay, B.; Bayrakdaroglu, A. An empirical research on investor biases in financial decision-making, financial risk tolerance and financial personality. *Int. J. Financial Res.* **2016**, *7*, 171. [CrossRef]
126. Lim, L.-H.; Benbasat, I. The debiasing role of group support systems: An experimental investigation of the representativeness bias. *Int. J. Human-Computer Stud.* **1997**, *47*, 453–471. [CrossRef]
127. Zhang, G.; Zhu, A.-X. The representativeness and spatial bias of volunteered geographic information: A review. *Ann. GIS* **2018**, *24*, 151–162. [CrossRef]
128. Howard, J. Representativeness Bias. In *Cognitive Errors and Diagnostic Mistakes: A Case-Based Guide to Critical Thinking in Medicine*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 425–443.
129. Park, H.; Yoo, Y. A literature review on chief executive officer hubris and related constructs: Is the theory of chief executive officer hubris an antecedents or consequences? *J. Appl. Bus. Res. (JABR)* **2017**, *33*, 703. [CrossRef]
130. Sadler-Smith, E. Hubris in business and management research: A 30-year review of studies. *Intox. Power Interdiscip. Insights* **2016**, 39–74.
131. Sadler-Smith, E.; Akstinaite, V.; Robinson, G.; Wray, T. Hubristic leadership: A review. *Leadership* **2017**, *13*, 525–548. [CrossRef]
132. Picone, P.M.; Galvagno, M.; Pisano, V. Hubris research in business: Taking stock and moving forward. *Manag. Decis.* **2024**, *62*, 1–24. [CrossRef]
133. Moritz, S.; Göritz, A.S.; Gallinat, J.; Schafschetzy, M.; Van Quaquebeke, N.; Peters, M.J.; Andreou, C. Subjective competence breeds overconfidence in errors in psychosis. A hubris account of paranoia. *J. Behav. Ther. Exp. Psychiatry* **2015**, *48*, 118–124. [CrossRef]
134. Brennan, N.M.; Conroy, J.P. Executive hubris: The case of a bank CEO. *Accounting, Audit. Account. J.* **2013**, *26*, 172–195. [CrossRef]
135. Gavade, R.K. Multi-Criteria Decision Making: An overview of different selection problems and methods. *Int. J. Comput. Sci. Inf. Technol.* **2014**, *5*, 5643–5646.
136. Balayn, A.; Lofi, C.; Houben, G.-J. Managing bias and unfairness in data for decision support: A survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems. *VLDB J.* **2021**, *30*, 739–768. [CrossRef]
137. Reddy, G.T.; Reddy, M.P.K.; Lakshmanna, K.; Kaluri, R.; Rajput, D.S.; Srivastava, G.; Baker, T. Analysis of Dimensionality Reduction Techniques on Big Data. *IEEE Access* **2020**, *8*, 54776–54788. [CrossRef]
138. Triantaphyllou, E. *Multi-Criteria Decision Making Methods*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 5–21. [CrossRef]
139. Hariri, R.H.; Fredericks, E.M.; Bowers, K.M. Uncertainty in big data analytics: Survey, opportunities, and challenges. *J. Big Data* **2019**, *6*, 44. [CrossRef]
140. Norori, N.; Hu, Q.; Aellen, F.M.; Faraci, F.D.; Tzovara, A. Addressing bias in big data and AI for health care: A call for open science. *Patterns* **2021**, *2*, 100347. [CrossRef]
141. Triguero, I.; Peralta, D.; Bacardit, J.; García, S.; Herrera, F. MRPR: A MapReduce solution for prototype reduction in big data classification. *Neurocomputing* **2015**, *150*, 331–345. [CrossRef]
142. Edwards, J.S.; Rodriguez, E. Remedies against bias in analytics systems. *J. Bus. Anal.* **2019**, *2*, 74–87. [CrossRef]
143. Cai, G.; Mahadevan, S. Big Data Analytics in Uncertainty Quantification: Application to Structural Diagnosis and Prognosis. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part A Civ. Eng.* **2018**, *4*, 949. [CrossRef]
144. Greenhalgh, T. Intuition and evidence--uneasy bedfellows? *Br. J. Gen. Pract.* **2002**, *52*, 395–400.
145. Gaudiano, B.A.; Brown, L.A.; Miller, I.W. Let your intuition be your guide? Individual differences in the evidence-based practice attitudes of psychotherapists. *J. Evaluation Clin. Pr.* **2011**, *17*, 628–634. [CrossRef] [PubMed]
146. Sheth, S.A.; Kwon, C.-S.; Barker, F.G. The Art of Management Decision Making: From Intuition to Evidence-based Medicine. *Otolaryngol. Clin. N. Am.* **2012**, *45*, 333–351. [CrossRef] [PubMed]
147. Welsh, I.; Lyons, C.M. Evidence-based care and the case for intuition and tacit knowledge in clinical assessment and decision making in mental health nursing practice: An empirical contribution to the debate. *J. Psychiatr. Ment. Health Nurs.* **2001**, *8*, 299–305. [CrossRef]
148. Chilcote, D.R. Intuition: A Concept Analysis. *Nurs. Forum* **2017**, *52*, 62–67. [CrossRef]
149. Adam, F.; Dempsey, E. Intuition in decision making Risk and opportunity. *J. Decis. Syst.* **2020**, *29*, 98–116. [CrossRef]
150. Hamilton, D.K. Evidence, Intuition, and Experiment: Partners in the Design Process. *HERD Health Environ. Res. Des. J.* **2019**, *12*, 66–71. [CrossRef]
151. Taylor, N.; McKay, S.; Long, J.C.; Gaff, C.; North, K.; Braithwaite, J.; Francis, J.J.; Best, S. Aligning intuition and theory: A novel approach to identifying the determinants of behaviours necessary to support implementation of evidence into practice. *Implement. Sci.* **2023**, *18*, 1–17. [CrossRef]

152. Braude, H.D. Clinical intuition versus statistics: Different modes of tacit knowledge in clinical epidemiology and evidence-based medicine. *Theor. Med. Bioeth.* **2009**, *30*, 181–198. [CrossRef] [PubMed]
153. Balaji, S. Role of research evidence in clinical decision–making: Intuition versus clinical experience versus scientific evidence. *Indian J. Dent. Res.* **2018**, *29*, 543–544. [CrossRef] [PubMed]
154. Vanlommel, K.; Van Gasse, R.; Vanhoof, J.; Van Petegem, P. Teachers' decision-making: Data based or intuition driven? *Int. J. Educ. Res.* **2017**, *83*, 75–83. [CrossRef]
155. Erisman, J.C.; de Sabbata, K.; Zuiderent-Jerak, T.; Syurina, E.V. Navigating complexity of child abuse through intuition and evidence-based guidelines: A mix-methods study among child and youth healthcare practitioners. *BMC Fam. Pr.* **2020**, *21*, 157. [CrossRef]
156. Liebowitz, J.; Chan, Y.; Jenkin, T.; Spicker, D.; Paliszkiewicz, J.; Babiloni, F. If numbers could "feel": How well do executives trust their intuition? *VINE J. Inf. Knowl. Manag. Syst.* **2019**, *49*, 531–545. [CrossRef]
157. Long, J.; Xie, L.; Wang, X.; Zhang, J.; Lu, B.; Wei, C.; Dai, D.; Zhu, G.; Tian, M. A Comprehensive Review of Signal Processing and Machine Learning Technologies for UHF PD Detection and Diagnosis (II): Pattern Recognition Approaches. *IEEE Access* **2024**, *12*, 29850–29890. [CrossRef]
158. Liao, H.-J.; Lin, C.-H.R.; Lin, Y.-C.; Tung, K.-Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
159. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Umar, A.M.; Linus, O.U.; Arshad, H.; Kazaure, A.A.; Gana, U.; Kiru, M.U. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access* **2019**, *7*, 158820–158846. [CrossRef]
160. Yu, D.; Zhang, Y.; Chen, Z. A comprehensive approach to the recovery of design pattern instances based on sub-patterns and method signatures. *J. Syst. Softw.* **2015**, *103*, 1–16. [CrossRef]
161. Schmidl, S.; Wenig, P.; Papenbrock, T. Anomaly detection in time series: A comprehensive evaluation. *Proc. Vldb Endow.* **2022**, *15*, 1779–1797. [CrossRef]
162. Saxena, A.; Khanna, A.; Gupta, D. Emotion recognition and detection methods: A comprehensive survey. *J. Artif. Intell. Syst.* **2020**, *2*, 53–79. [CrossRef]
163. Khezerlou, A.V.; Zhou, X.; Li, L.; Shafiq, Z.; Liu, A.X.; Zhang, F. A traffic flow approach to early detection of gathering events. *ACM Trans. Intell. Syst. Technol.* **2017**, *8*, 1–24. [CrossRef]
164. Florkowski, M. Anomaly detection, trend evolution, and feature extraction in partial discharge patterns. *Energies* **2021**, *14*, 3886. [CrossRef]
165. Cao, D.M.; Sayed, A.; Mia, T.; Ayon, E.H.; Ghosh, B.P.; Ray, R.K.; Raihan, A.; Akter, A.; Rahman, M.; Islam, T.; et al. Advanced cybercrime detection: A comprehensive study on supervised and unsupervised machine learning approaches using real-world datasets. *J. Comput. Sci. Technol. Stud.* **2024**, *6*, 40–48. [CrossRef]
166. Gupta, M.; Wadhvani, R.; Rasool, A. Comprehensive analysis of change-point dynamics detection in time series data: A review. *Expert Syst. Appl.* **2024**, *248*, 123342. [CrossRef]
167. Waghade, S.S.; Karandikar, A.M. A comprehensive study of healthcare fraud detection based on machine learning. *Int. J. Appl. Eng. Res.* **2018**, *13*, 4175–4178.
168. Nie, L.; Zhao, Y.; Li, C.; Luo, X.; Liu, Y. Shadows in the Interface: A Comprehensive Study on Dark Patterns. *Proc. ACM Softw. Eng.* **2024**, *1*, 204–225. [CrossRef]
169. Casarrubea, M.; Jonsson, G.; Faulisi, F.; Sorbera, F.; Di Giovanni, G.; Benigno, A.; Crescimanno, G.; Magnusson, M. T-pattern analysis for the study of temporal structure of animal and human behavior: A comprehensive review. *J. Neurosci. Methods* **2015**, *239*, 34–46. [CrossRef]
170. Chen, C.; Xie, Y.; Lin, S.; Yao, A.; Jiang, G.; Zhang, W.; Qu, Y.; Qiao, R.; Ren, B.; Ma, L. Comprehensive Regularization in a Bi-directional Predictive Network for Video Anomaly Detection. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 230–238. [CrossRef]
171. Marwala, T. Filtering Irrelevant Information for Rational Decision Making. *Artif. Intell. Tech. Ration. Decis. Mak.* **2014**, 111–130.
172. Nasr, S.; Moeeny, A.; Esteky, H. Neural correlate of filtering of irrelevant information from visual working memory. *PLoS ONE* **2008**, *3*, e3282. [CrossRef]
173. Gupta, S.; Wang, H.; Lipton, Z.; Wang, Y. Correcting exposure bias for link recommendation. In Proceedings of the 38 th International Conference on Machine Learning, PMLR 139, 2021., Virtual, 18–24 July 2021; Available online: https://proceedings.mlr.press/v139/gupta21c.html (accessed on 30 September 2025).
174. Chen, L.; Fang, B.; Shang, Z.; Tang, Y. Negative samples reduction in cross-company software defects prediction. *Inf. Softw. Technol.* **2015**, *62*, 67–77. [CrossRef]
175. Ran, A.R.; Shi, J.; Ngai, A.K.; Chan, W.-Y.; Chan, P.P.; Young, A.L.; Yung, H.-W.; Tham, C.C.; Cheung, C.Y. Artificial intelligence deep learning algorithm for discriminating ungradable optical coherence tomography three-dimensional volumetric optic disc scans. *Neurophotonics* **2019**, *6*, 041110. [CrossRef] [PubMed]

176. Green, A.; Wright, M. Reduction of task-relevant information in skill acquisition. *Eur. J. Cogn. Psychol.* **2003**, *15*, 267–290. [CrossRef]
177. Turrentine, F.E.; Dreisbach, C.N.; Ivany, A.R.S.; Hanks, J.B.; Schroen, A.T. Influence of gender on surgical residency applicants' recommendation letters. *J. Am. Coll. Surg.* **2019**, *228*, 356–365e3. [CrossRef]
178. Lenarduzzi, V.; Dieste, O.; Fucci, D.; Vegas, S. Towards a methodology for participant selection in software engineering experiments: A vision of the future. In Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Bari, Italy, 11–15 October 2021; pp. 1–6.
179. Thabtah, F.; Kamalov, F.; Hammoud, S.; Shahamiri, S.R. Least Loss: A simplified filter method for feature selection. *Inf. Sci.* **2020**, *534*, 1–15. [CrossRef]
180. Ravindranath, V.; Ramasamy, S.; Somula, R.; Sahoo, K.S.; Gandomi, A.H. Swarm Intelligence Based Feature Selection for Intrusion and Detection System in Cloud Infrastructure. In Proceedings of the 2020 IEEE Congress on EVOLUTIONARY Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–6.
181. Ayalon, O.; Toch, E. Retrospective privacy: Managing longitudinal privacy in online social networks. In Proceedings of the Ninth Symposium on Usable Privacy and Security, Newcastle, UK, 24–26 July 2013; pp. 1–13.
182. Bhat, N.A.; Farooq, S.U. An improved method for training data selection for cross-project defect prediction. *Arab. J. Sci. Eng.* **2022**, *47*, 1939–1954. [CrossRef]
183. Elouataoui, W. AI-Driven frameworks for enhancing data quality in big data ecosystems: Error_detection, correction, and metadata integration. *arXiv* **2024**, arXiv:2405.03870.
184. Nahmias, Y.; Perel, M. The oversight of content moderation by AI: Impact assessments and their limitations. *Harv. J. Legis.* **2021**, *58*, 145.
185. Rallabandi, S.; Kakodkar, I.G.S.; Avuku, O. Ethical U se of AI in Social Media. In Proceedings of the 2023 International Workshop on Intelligent Systems (IWIS), Ulsan, Republic of Korea, 9–11 August 2023; pp. 1–9.
186. Kerdvibulvech, C. Big data and AI-driven evidence analysis: A global perspective on citation trends, accessibility, and future research in legal applications. *J. Big Data* **2024**, *11*, 1–24. [CrossRef]
187. Hey, J.D. Does repetition improve consistency? *Exp. Econ.* **2001**, *4*, 5–54. [CrossRef]
188. Francis, G. Replication, statistical consistency, and publication bias. *J. Math. Psychol.* **2013**, *57*, 153–169. [CrossRef]
189. Begg, I.; Armour, V. Repetition and the ring of truth: Biasing comments. *Can. J. Behav. Sci. / Rev. Can. des Sci. du Comport.* **1991**, *23*, 195–213. [CrossRef]
190. Schulz-Hardt, S.; Giersiepen, A.; Mojzisch, A. Preference-consistent information repetitions during discussion: Do they affect subsequent judgments and decisions? *J. Exp. Soc. Psychol.* **2016**, *64*, 41–49. [CrossRef]
191. Guan, J.; Huang, M. Mitigating the Learning Bias towards Repetition by Self-Contrastive Training for Open-Ended Generation. *arXiv* **2023**, arXiv:2307.01542.
192. Bureš, V.; Ponce, D.; Čech, P.; Mls, K. The effect of trial repetition and problem size on the consistency of decision making. *PLoS ONE* **2019**, *14*, e0216235. [CrossRef]
193. Shi, W.; Li, Y.; Sahay, S.; Yu, Z. Refine and imitate: Reducing repetition and inconsistency in persuasion dialogues via reinforcement learning and human demonstration. *arXiv* **2020**, arXiv:2012.15375.
194. Sun, R.; Gong, Z.; Gao, G.; Shah, A.A. Comparative analysis of Multi-Criteria Decision-Making methods for flood disaster risk in the Yangtze River Delta. *Int. J. Disaster Risk Reduct.* **2020**, *51*, 101768. [CrossRef]
195. Russo, J.E.; Carlson, K.A.; Meloy, M.G.; Yong, K. The goal of consistency as a cause of information distortion. *J. Exp. Psychol. Gen.* **2008**, *137*, 456–470. [CrossRef]
196. Huang, H.-H.; Kuo, M.-C. Looks familiar, appears more valid? The moderating role of computer-supported warnings between information repetition and decision outcome. *Behav. Inf. Technol.* **2015**, *34*, 1119–1128. [CrossRef]
197. Hunzaker, M.B.F. Cultural sentiments and schema-consistency bias in information transmission. *Am. Sociol. Rev.* **2016**, *81*, 1223–1250. [CrossRef]
198. Deng, Z.; Li, J.; Liu, W.; Zhao, J. Unbiased Interest Modeling in Sequential Basket Analysis: Addressing Repetition Bias with Multi-Factor Estimation. *ACM Trans. Recomm. Syst.* **2025**, *3*, 1–27. [CrossRef]
199. Powers, D.M.W.; Atyabi, A. The problem of cross-validation: Averaging and bias, repetition and significance. In Proceedings of the 2012 Spring Congress on Engineering and Technology, Xi'an, China, 27–30 May 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–5.
200. Oyeniran, O.C.; Adewusi, A.O.; Adeleke, A.G.; Akwawa, L.A.; Azubuko, C.F. Ethical AI: Addressing bias in machine learning models and software applications. *Comput. Sci. IT Res. J.* **2022**, *3*, 115–126. [CrossRef]
201. Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv.* **2021**, *54*, 1–35. [CrossRef]
202. Siddique, S.; Haque, M.A.; George, R.; Gupta, K.D.; Gupta, D.; Faruk, J.H. Survey on machine learning biases and mitigation techniques. *Digital* **2023**, *4*, 1–68. [CrossRef]

203. Hanna, M.G.; Pantanowitz, L.; Jackson, B.; Palmer, O.; Visweswaran, S.; Pantanowitz, J.; Deebajah, M.; Rashidi, H.H. Ethical and bias considerations in artificial intelligence/machine learning. *Mod. Pathol.* **2024**, *38*, 100686. [CrossRef]
204. Lee, N.T. Detecting racial bias in algorithms and machine learning. *J. Inf. Commun. Ethic. Soc.* **2018**, *16*, 252–260. [CrossRef]
205. Oguntibeju, O.O. Mitigating artificial intelligence bias in financial systems: A comparative analysis of debiasing techniques. *Asian J. Res. Comput. Sci.* **2024**, *17*, 165–178. [CrossRef]
206. Zheng, W.; Chung, L.M.; Zhao, H. Bias detection and correction in RNA-Sequencing data. *BMC Bioinform.* **2011**, *12*, 290. [CrossRef]
207. Courchamp, F.; Say, L.; Pontier, D. Detection, identification, and correction of a bias in an epidemiological study. *J. Wildl. Dis.* **2000**, *36*, 71–78. [CrossRef]
208. Zheng, W.X. A bias correction method for identification of linear dynamic errors-in-variables models. *IEEE Trans. Autom. Control.* **2002**, *47*, 1142–1147. [CrossRef]
209. Haladyna, T.; Hess, R.K. The detection and correction of bias in student ratings of instruction. *Res. High. Educ.* **1994**, *35*, 669–687. [CrossRef]
210. Sadker, D.M.; Zittleman, K. Practical strategies for detecting and correcting gender bias in your classroom. *Gend. Classr. Found. Ski. Methods Strateg. Across Curric.* **2007**, 259–275.
211. Hofmeester, T.R.; Cromsigt, J.P.G.M.; Odden, J.; Andrén, H.; Kindberg, J.; Linnell, J.D.C. Framing pictures: A conceptual framework to identify and correct for biases in detection probability of camera traps enabling multi-species comparison. *Ecol. Evol.* **2019**, *9*, 2320–2336. [CrossRef] [PubMed]
212. Elhaddad, M.; Hamam, S. AI-driven clinical decision support systems: An ongoing pursuit of potential. *Cureus* **2024**, *16*, e57728. [CrossRef]
213. Narne, S.; Adedoja, T.; Mohan, M.; Ayyalasomayajula, T. AI-driven decision support systems in management: Enhancing strategic planning and execution. *Int. J. Recent Innov. Trends Comput. Commun.* **2024**, *12*, 268–276.
214. Niranjan, K.; Kumar, S.S.; Vedanth, S.; Chitrakala, S. An Explainable AI driven Decision Support System for COVID-19 Diagnosis using Fused Classification and Segmentation. *Procedia Comput. Sci.* **2023**, *218*, 1915–1925. [CrossRef]
215. Braun, M.; Hummel, P.; Beck, S.; Dabrock, P. Primer on an ethics of AI-based decision support systems in the clinic. *J. Med. Ethic.* **2020**, *47*, e3. [CrossRef]
216. Dlugatch, R.; Georgieva, A.; Kerasidou, A. AI-driven decision support systems and epistemic reliance: A qualitative study on obstetricians' and midwives' perspectives on integrating AI-driven CTG into clinical decision making. *BMC Med. Ethic.* **2024**, *25*, 6. [CrossRef]
217. Ekundayo, F. Leveraging Ai-Driven Decision Intelligence For Complex Systems Engineering. *Int. J. Res. Publ. Rev.* **2024**, *5*, 5489–5499. [CrossRef]
218. Almalki, S.S. AI-Driven Decision Support Systems in Agile Software Project Management: Enhancing Risk Mitigation and Resource Allocation. *Systems* **2025**, *13*, 208. [CrossRef]
219. Mohammed, A.S.; Neravetla, A.R.; Nomula, V.K.; Gupta, K.; Dhanasekaran, S. Understanding the Impact of AI-driven Clinical Decision Support Systems. In Proceedings of the 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 24–28 June 2024; pp. 1–6.
220. Bleher, H.; Braun, M. Diffused responsibility: Attributions of responsibility in the use of AI-driven clinical decision support systems. *AI Ethic.* **2022**, *2*, 747–761. [CrossRef]
221. Martínez-Cid, S.; Vallejo, D.; Herrera, V.; Schez-Sobrino, S.; Castro-Schez, J.J.; Albusac, J.A. Explainable AI-driven decision support system for personalizing rehabilitation routines in stroke recovery. *Prog. Artif. Intell.* **2025**, 1–23. [CrossRef]
222. Comito, C.; Falcone, D.; Forestiero, A. AI-driven clinical decision support: Enhancing disease diagnosis exploiting patients similarity. *IEEE Access* **2022**, *10*, 6878–6888. [CrossRef]
223. Leschanowsky, A.; Popp, B.; Peters, N. Debiasing strategies for conversational ai: Improving privacy and security decision-making. *Digit. Soc.* **2023**, *2*, 34. [CrossRef]
224. Salvetti, F.; Bertagni, B.; Contardo, I. Cognitive Clarity: Learning Unbiased Decision-Making and Critical Thinking from AI. In *Creative Approaches to Technology-Enhanced Learning for the Workplace and Higher Education: Proceedings of 'The Learning Ideas Conference'2024*; Springer Nature: London, UK, 2024; Volume 2, p. 114.
225. Bouchagiar, G. The Long Road Toward Tracking the Trackers and De-biasing: A Consensus on Shaking the Black Box and Freeing From Bias. *Rev. Eur. Stud.* **2019**, *11*, 27. [CrossRef]
226. McHugh, A.P.; Smith, J.L.; Sieck, W.R. Of Collaborative Decision Making. *Nat. Decis. Mak. Mac-Rocognition* **2008**, *141*.
227. Wilson, C.G.; Bond, C.E.; Shipley, T.F. How can geologic decision-making under uncertainty be improved? *Solid Earth* **2019**, *10*, 1469–1488. [CrossRef]
228. Stone, M.; Aravopoulou, E.; Ekinci, Y.; Evans, G.; Hobbs, M.; Labib, A.; Laughlin, P.; Machtynger, J.; Machtynger, L. Artificial intelligence (AI) in strategic marketing decision-making: A research agenda. *Bottom Line* **2020**, *33*, 183–200. [CrossRef]
229. Churaman, B. Gout or Deep Vein Thrombosis: Expanding Differential Diagnosis and Avoiding Anchoring Bias. *J. Nurse Pr.* **2024**, *20*, 105216. [CrossRef]

230. Shrestha, Y.R.; Ben-Menahem, S.M.; von Krogh, G. Organizational decision-making structures in the age of artificial intelligence. *Calif. Manag. Rev.* **2019**, *61*, 66–83. [CrossRef]
231. Gupta, A.; Carpenter, D.; Min, W.; Rowe, J.; Azevedo, R.; Lester, J. Detecting and mitigating encoded bias in deep learning-based stealth assessment models for reflection-enriched game-based learning environments. *Int. J. Artif. Intell. Educ.* **2023**, *34*, 1138–1165. [CrossRef]
232. Pareek, C.S. Unmasking bias: A Framework for Testing and Mitigating AI Bias in Insurance Underwriting Models. *J. Artif. Intell. Mach. Learn. Data Sci.* **2023**, *1*, 1736–1741. [CrossRef]
233. Schmitt, M. Automated machine learning: AI-driven decision making in business analytics. *Intell. Syst. Appl.* **2023**, *18*, 200188. [CrossRef]
234. Abbas, F. Evaluating the Performance of AI-Driven Systems in Real-World Applications. *Artif. Intell. Multi-Discip. J. Syst. Appl.* **2024**, *1*, 18–26.
235. Selvarajan, G. Leveraging AI-Enhanced Analytics for Industry-Specific Optimization: A Strategic Approach to Transforming Data-Driven Decision-Making. *Int. J. Enhanc. Res. Sci. Technol. Eng.* **2021**, *10*, 78–84.
236. Burstein, F.; Holsapple, C.W.; Power, D.J. Decision support systems: A historical overview. In *Handbook on Decision Support Systems 1: Basic Themes*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 121–140.
237. Ebadi, A.; Xi, P.; MacLean, A.; Florea, A.; Tremblay, S.; Kohli, S.; Wong, A. COVIDx-US: An open-access benchmark dataset of ultrasound imaging data for AI-driven COVID-19 analytics. *Front. Biosci.* **2021**, *27*, 198. [CrossRef]
238. Shukla, R.; Sinha, A.; Chaudhary, A. TweezBot: An AI-driven online media bot identification algorithm for twitter social networks. *Electronics* **2022**, *11*, 743. [CrossRef]
239. Olutimehin, A.T.; Ajayi, A.J.; Metibemu, O.C.; Balogun, A.Y.; Oladoyinbo, T.O.; Olaniyi, O.O. Adversarial threats to AI-driven systems: Exploring the attack surface of machine learning models and countermeasures. *J. Eng. Res. Rep.* **2025**, *27*, 341–362. [CrossRef]
240. Yaiprasert, C.; Hidayanto, A.N. AI-driven ensemble three machine learning to enhance digital marketing strategies in the food delivery business. *Intell. Syst. Appl.* **2023**, *18*, 200235. [CrossRef]
241. Tufano, M.; Agarwal, A.; Jang, J.; Moghaddam, R.Z.; Sundaresan, N. AutoDev: Automated AI-driven development. *arXiv* **2024**, arXiv:2403.08299.
242. Bhowmik, A.; Sannigrahi, M.; Chowdhury, D.; Dey, A.; Gill, S.S. CloudAISim: A toolkit for modelling and simulation of modern applications in AI-driven cloud computing environments. *BenchCouncil Trans. Benchmarks Stand. Eval.* **2023**, *3*, 100150. [CrossRef]
243. Liu, Y.; Wang, J. AI-Driven Health Advice: Evaluating the Potential of Large Language Models as Health Assistants. *J. Comput. Methods Eng. Appl.* **2023**, 1–7. [CrossRef]
244. Terziyan, V.; Gryshko, S.; Golovianko, M. Patented intelligence: Cloning human decision models for Industry 4.0. *J. Manuf. Syst.* **2018**, *48*, 204–217. [CrossRef]
245. East, R. Bias in the evaluation of research methods. *Mark. Theory* **2016**, *16*, 219–231. [CrossRef]
246. Crain, R.L.; York, R.L. Evaluating a Successful Program: Experimental Method and Academic Bias. *Sch. Rev.* **1976**, *84*, 233–254. [CrossRef]
247. Wu, J.J.; Mazzuchi, T.A.; Sarkani, S. Comparison of multi-criteria decision-making methods for online controlled experiments in a launch decision-making framework. *Inf. Softw. Technol.* **2023**, *155*, 107115. [CrossRef]
248. Kamalbasha, S.; Eugster, M.J.A. Bayesian A/B testing for business decisions. In Proceedings of the International Data Science Conference, Hong Kong, China, 23–25 May 2020; Springer Fachmedien: Wiesbaden, Germany, 2020; pp. 50–57.
249. Quin, F.; Weyns, D.; Galster, M.; Silva, C.C. A/B testing: A systematic literature review. *J. Syst. Softw.* **2024**, *211*, 112011. [CrossRef]
250. Robinson, S.; Alifantis, T.; Edwards, J.S.; Ladbrook, J.; Waller, A. Knowledge-based improvement: Simulation and artificial intelligence for identifying and improving human decision-making in an operations system. *J. Oper. Res. Soc.* **2005**, *56*, 912–921. [CrossRef]
251. Dai, C.-P.; Ke, F. Educational applications of artificial intelligence in simulation-based learning: A systematic mapping review. *Comput. Educ. Artif. Intell.* **2022**, *3*, 100087. [CrossRef]
252. Ivezic, N.; Garrett, J.H. Machine learning for simulation-based support of early collaborative design. *Artif. Intell. Eng. Des. Anal. Manuf.* **1998**, *12*, 123–139. [CrossRef]
253. Çubukçu, H.C.; Topcu, D.I.; Yenice, S. Machine learning-based clinical decision support using laboratory data. *Clin. Chem. Lab. Med. (CCLM)* **2024**, *62*, 793–823. [CrossRef]
254. Hoffmann, G.E. Concepts for the third generation of laboratory systems. *Clin. Chim. Acta* **1998**, *278*, 203–216. [CrossRef]
255. van Rossum, H.H. Technical quality assurance and quality control for medical laboratories: A review and proposal of a new concept to obtain integrated and validated QA/QC plans. *Crit. Rev. Clin. Lab. Sci.* **2022**, *59*, 586–600. [CrossRef]
256. Lai, V.; Chen, C.; Liao, Q.V.; Smith-Renner, A.; Tan, C. Towards a science of human-ai decision making: A survey of empirical studies. *arXiv* **2021**, arXiv:2112.11471. [CrossRef]

257. Vereschak, O.; Bailly, G.; Caramiaux, B. How to evaluate trust in AI-assisted decision making? A survey of empirical methodologies. *Proc. ACM Human-Computer Interact.* **2021**, *5*, 1–39. [CrossRef]
258. Hussain, M.J. A Survey Based on Behavior Analysis of Artificial Intelligence Using Machine Learning Process. In Proceedings of the 2024 4th International Conference on Sustainable Expert Systems (ICSES), Lekhnath, Nepal, 15–17 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1694–1701.
259. Phillips-Wren, G. AI tools in decision making support systems: A review. *Int. J. Artif. Intell. Tools* **2012**, *21*, 1240005. [CrossRef]
260. Zlaugotne, B.; Zihare, L.; Balode, L.; Kalnbalkite, A.; Khabdullin, A.; Blumberga, D. Multi-criteria decision analysis methods comparison. *Sci. J. Riga Tech. Univ. Environ. Clim. Technol.* **2020**, *24*, 454–471. [CrossRef]
261. Rachman, A.P.; Ichwania, C.; Mangkuto, R.A.; Pradipta, J.; Koerniawan, M.D.; Sarwono, J. Comparison of multi-criteria decision-making methods for selection of optimum passive design strategy. *Energy Build.* **2024**, *314*, 114285. [CrossRef]
262. Kohavi, R.; Longbotham, R. Online controlled experiments and A/B tests. In *Encyclopedia of Machine Learning and Data Science*; Springer: New York, NY, USA, 2015; pp. 1–11.
263. Wan, R.; Liu, Y.; McQueen, J.; Hains, D.; Song, R. Experimentation platforms meet reinforcement learning: Bayesian sequential decision-making for continuous monitoring. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 5016–5027.
264. Hall, T.A.; Hasan, S. Organizational decision-making and the returns to experimentation. *J. Organ. Des.* **2022**, *11*, 129–144. [CrossRef]
265. Koop, G.J.; Johnson, J.G. Response dynamics: A new window on the decision process. *Judgm. Decis. Mak.* **2011**, *6*, 750–758. [CrossRef]
266. Foss, D.J. Decision processes during sentence comprehension: Effects of lexical item difficulty and position upon decision times. *J. Verbal Learn. Verbal Behav.* **1969**, *8*, 457–462. [CrossRef]
267. Winter, G.; Lobley, C.M.C.; Prince, S.M. Decision making in*xia*2. *Acta Crystallogr. Sect. D Struct. Biol.* **2013**, *69*, 1260–1273. [CrossRef]
268. Ke, Y.; Yang, R.; Lie, S.A.; Lim, T.X.Y.; Ning, Y.; Li, I.; Abdullah, H.R.; Ting, D.S.W.; Liu, N. Mitigating cognitive biases in clinical decision-making through multi-agent conversations using large language models: Simulation study. *J. Med. Internet Res.* **2024**, *26*, e59439. [CrossRef]
269. Hagendorff, T.; Fabi, S. Why we need biased AI: How including cognitive biases can enhance AI systems. *J. Exp. Theor. Artif. Intell.* **2023**, *36*, 1885–1898. [CrossRef]
270. Hao, X.; Demir, E.; Eyers, D. Exploring collaborative decision-making: A quasi-experimental study of human and Generative AI interaction. *Technol. Soc.* **2024**, *78*, 102662. [CrossRef]
271. Cant, R.P.; Cooper, S.J. Simulation-based learning in nurse education: Systematic review. *J. Adv. Nurs.* **2009**, *66*, 3–15. [CrossRef] [PubMed]
272. Akter, S.; Sultana, S.; Mariani, M.; Wamba, S.F.; Spanaki, K.; Dwivedi, Y.K. Advancing algorithmic bias management capabilities in AI-driven marketing analytics research. *Ind. Mark. Manag.* **2023**, *114*, 243–261. [CrossRef]
273. Lavin, A.; Krakauer, D.; Zenil, H.; Gottschlich, J.; Mattson, T.; Brehmer, J.; Anandkumar, A.; Choudry, S.; Rocki, K.; Baydin, A.G.; et al. Simulation intelligence: Towards a new generation of scientific methods. *arXiv* **2021**, arXiv:2112.03235.
274. AlMakinah, R.; Goodarzi, M.; Tok, B.; Canbaz, M.A. Mapping artificial intelligence bias: A network-based framework for analysis and mitigation. *AI Ethic.* **2024**, *5*, 1995–2014. [CrossRef]
275. Minotra, D.; Feigh, K. Reviewing linkages between display design and cognitive biases in decision making: An emergency response perspective. *Theor. Issues Ergon. Sci.* **2024**, *25*, 776–803. [CrossRef]
276. Küper, A.; Lodde, G.; Livingstone, E.; Schadendorf, D.; Krämer, N. Mitigating cognitive bias with clinical decision support systems: An experimental study. *J. Decis. Syst.* **2023**, *33*, 439–458. [CrossRef]
277. Bojić, L.; Cinelli, M.; Ćulibrk, D.; Delibašić, B. CERN for AI: A theoretical framework for autonomous simulation-based artificial intelligence testing and alignment. *Eur. J. Futur. Res.* **2024**, *12*, 1–19. [CrossRef]
278. Leavy, S. Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning. In Proceedings of the 1st International Workshop on Gender Equality in Software Engineering, Gothenburg, Sweden, 27 May–3 June 2018; pp. 14–16.
279. Ferreira, R.; Gomes, J.S.; Carvalho, A. Absorptive capacity in analytics implementations: A research model. *RAM* **2020**, *21*, eRAMR200036. [CrossRef]
280. Joshi, K.; Yong, J.; Tao, X.; Zhang, Y. Big data in healthcare has improved the care quality! A reality or myth? In Proceedings of the 2024 Twelfth International Conference on Advanced Cloud and Big Data (CBD), Brisbane, Australia, 28 November–2 December 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 333–338.
281. Zschech, P.; Benedict, M.I. Paper 8-Towards Maintenance Analytics Ecosystems. *Des. Eval. Do-Main-Specif. Platf. Spec. Case Digit. Healthc.* **2020**, 99.

282. Intezari, A.; Namvar, M.; Taghinejad, R. Knowledge identity (KI): A determining factor in the effective use of analytics. *Knowl. Manag. Res. Pr.* **2022**, *20*, 14–33. [CrossRef]
283. Shamsuddin, S.M.; Hasan, S. Data science vs big data@ UTM big data centre. In Proceedings of the 2015 International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 27–28 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–4.
284. Bo, C.A.; Alejandro, L.L.; Calzado, J.K.R.; Campaña, E.J.G.; Rodriguez, C.J.D. Patient Records Management System With Data Analytics for Singapore Diagnostics. In Proceedings of the 2023 IEEE 21st Student Conference on Research and Development (SCOReD), Kuala Lumpur, Malaysia, 13–14 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 585–590.
285. Krleza, J.L.; Honovic, L.; Tanaskovic, J.V.; Podolar, S.; Rimac, V.; Jokic, A. Post-analytical laboratory work: National recommendations from the Working Group for Post-analytics on behalf of the Croatian Society of Medical Biochemistry and Laboratory Medicine. *Biochem. Medica* **2019**, *29*, 228–261. [CrossRef]
286. Jokic, A.; Tanaskovic, J.V.; Honovic, L.; Krleza, J.L.; Podolar, S.; Rimac, V. Interpretative comments need for harmonization? Results of the Croatian survey by the Working Group for Post-analytics. *Biochem. Medica* **2022**, *32*, 106–110. [CrossRef] [PubMed]
287. Mansoor, I.; Dar, F.J. Utilizing Data Analytics And Business Intelligence Tools In Laboratory Workflow. *EJIFCC* **2024**, *35*, 34–43.
288. Lou, A.H.; Elnenaei, M.O.; Sadek, I.; Thompson, S.; Crocker, B.D.; Nassar, B. Evaluation of the impact of a total automation system in a large core laboratory on turnaround time. *Clin. Biochem.* **2016**, *49*, 1254–1258. [CrossRef]
289. Chinta, S.V.; Wang, Z.; Palikhe, A.; Zhang, X.; Kashif, A.; Smith, M.A.; Liu, J.; Zhang, W. AI-driven healthcare: A review on ensuring fairness and mitigating bias. *PLOS Digit. Health* **2024**, *4*, e0000864. [CrossRef]
290. Sarkar, B.; Paul, R.K. Ethical AI and bias mitigation solution strategy in AI implementation. *J. ID* **2023**, *2157*, 0178.
291. Oliver, A.C.; Pratt, L.A.; Normand, M.P. A survey of functional behavior assessment methods used by behavior analysts in practice. *J. Appl. Behav. Anal.* **2015**, *48*, 817–829. [CrossRef] [PubMed]
292. Couvillon, M.A.; Bullock, L.M.; Gable, R.A. Tracking behavior assessment methodology and support strategies: A national survey of how schools utilize functional behavioral assessments and behavior intervention plans. *Emot. Behav. Difficulties* **2009**, *14*, 215–228. [CrossRef]
293. Graff, R.B.; Karsten, A.M. Assessing preferences of individuals with developmental disabilities: A survey of current practices. *Behav. Anal. Pr.* **2012**, *5*, 37–48. [CrossRef]
294. Nelson, J.R.; Roberts, M.L.; Rutherford Jr, R.B.; Mathur, S.R.; Aaroe, L.A. A statewide survey of special education administrators and school psychologists regarding functional behavioral assessment. *Educ. Treat. Child.* **1999**, 267–279.
295. Catania, J.A.; Moskowitz, J.T.; Ruiz, M.; Cleland, J. A review of national AIDS-related behavioral surveys. *AIDS* **1996**, *10*, S183–S190. [CrossRef]
296. Scott, T.M.; Meers, D.T.; Nelson, C.M. Toward a consensus of functional behavioral assessment for students with mild disabilities in public school contexts: A national survey. *Educ. Treat. Child.* **2000**, 265–285.
297. Barberis, N.; Thaler, R. A survey of behavioral finance. In *Handbook of The Economics of Finance*; Elsevier: Amsterdam, The Netherlands, 2003; Volume 1, pp. 1053–1128. [CrossRef]
298. Johnston, L.G.; Hakim, A.J.; Dittrich, S.; Burnett, J.; Kim, E.; White, R.G. A systematic review of published respondent-driven sampling surveys collecting behavioral and biologic data. *AIDS Behav.* **2016**, *20*, 1754–1776. [CrossRef] [PubMed]
299. Locher, F.M.; Philipp, M. Measuring reading behavior in large-scale assessments and surveys. *Front. Psychol.* **2023**, *13*, 1044290. [CrossRef]
300. Ellingson, S.A.; Miltenberger, R.G.; Long, E.S. A survey of the use of functional assessment procedures in agencies serving individuals with developmental disabilities. *Behav. Interv.* **1999**, *14*, 187–198. [CrossRef]
301. Vassakis, K.; Petrakis, E.; Kopanakis, I. Big data analytics: Applications, prospects and challenges. *Mob. Big Data A Roadmap Models Technol.* **2018**, 3–20.
302. Ahmed, E.; Yaqoob, I.; Hashem, I.A.T.; Khan, I.; Ahmed, A.I.A.; Imran, M.; Vasilakos, A.V. The role of big data analytics in Internet of Things. *Comput. Networks* **2017**, *129*, 459–471. [CrossRef]
303. Bhattarai, B.P.; Paudyal, S.; Luo, Y.; Mohanpurkar, M.; Cheung, K.; Tonkoski, R.; Hovsapian, R.; Myers, K.S.; Zhang, R.; Zhao, P.; et al. Big data analytics in smart grids: State-of-the-art, challenges, opportunities, and future directions. *IET Smart Grid* **2019**, *2*, 141–154. [CrossRef]
304. Janke, A.T.; Overbeek, D.L.; Kocher, K.E.; Levy, P.D. Exploring the potential of predictive analytics and big data in emergency care. *Ann. Emerg. Med.* **2016**, *67*, 227–236. [CrossRef]
305. Raghupathi, W.; Raghupathi, V. Big data analytics in healthcare: Promise and potential. *Health Inf. Sci. Syst.* **2014**, *2*, 3. [CrossRef]
306. Kuo, M.H.; Sahama, T.; Kushniruk, A.W.; Borycki, E.M.; Grunwell, D.K. Health big data analytics: Current perspectives, challenges and potential solutions. *Int. J. Big Data Intell.* **2014**, *1*, 114. [CrossRef]
307. Dubey, R.; Gunasekaran, A.; Childe, S.J.; Papadopoulos, T.; Luo, Z.; Wamba, S.F.; Roubaud, D. Can big data and predictive analytics improve social and environmental sustainability? *Technol. Forecast. Soc. Chang.* **2019**, *144*, 534–545. [CrossRef]

308. Olivera, P.; Danese, S.; Jay, N.; Natoli, G.; Peyrin-Biroulet, L. Big data in IBD: A look into the future. *Nat. Rev. Gastroenterol. Hepatol.* **2019**, *16*, 312–321. [CrossRef]
309. Mikalef, P.; Pappas, I.O.; Krogstie, J.; Giannakos, M. Big data analytics capabilities: A systematic literature review and research agenda. *Inf. Syst. e-Business Manag.* **2018**, *16*, 547–578. [CrossRef]
310. Günther, W.A.; Mehrizi, M.H.R.; Huysman, M.; Feldberg, F. Debating big data: A literature review on realizing value from big data. *J. Strat. Inf. Syst.* **2017**, *26*, 191–209. [CrossRef]
311. Shorfuzzaman, M.; Hossain, M.S.; Nazir, A.; Muhammad, G.; Alamri, A. Harnessing the power of big data analytics in the cloud to support learning analytics in mobile learning environment. *Comput. Hum. Behav.* **2019**, *92*, 578–588. [CrossRef]
312. Ram, J.; Zhang, C.; Koronios, A. The implications of big data analytics on business intelligence: A qualitative study in China. *Procedia Comput. Sci.* **2016**, *87*, 221–226. [CrossRef]
313. Griffin, G.P.; Mulhall, M.; Simek, C.; Riggs, W.W. Mitigating bias in big data for transportation. *J. Big Data Anal. Transp.* **2020**, *2*, 49–59. [CrossRef]
314. Favaretto, M.; De Clercq, E.; Elger, B.S. Big Data and discrimination: Perils, promises and solutions. A systematic review. *J. Big Data* **2019**, *6*, 12. [CrossRef]
315. Pagano, T.P.; Loureiro, R.B.; Lisboa, F.V.N.; Peixoto, R.M.; Guimarães, G.A.S.; Cruz, G.O.R.; Araujo, M.M.; Santos, L.L.; Cruz, M.A.S.; Oliveira, E.L.S.; et al. Bias and unfairness in machine learning models: A systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. *Big Data Cogn. Comput.* **2023**, *7*, 15. [CrossRef]
316. Khoei, T.T.; Singh, A. Data reduction in big data: A survey of methods, challenges and future directions. *Int. J. Data Sci. Anal.* **2024**, *20*, 1643–1682. [CrossRef]
317. Dung, H.T.; Do, D.T.; Nguyen, V.T. Comparison of multi-criteria decision making methods using the same data standardization method. *Strojnícky časopis-J. Mech. Eng.* **2022**, *72*, 57–72. [CrossRef]
318. Chen, P.; Wu, L.; Wang, L. AI fairness in data management and analytics: A review on challenges, methodologies and applications. *Appl. Sci.* **2023**, *13*, 10258. [CrossRef]
319. Schaffer, C. Overfitting Avoidance as Bias. *Mach. Learn.* **1993**, *10*, 153–178. [CrossRef]
320. Goddard, K.; Roudsari, A.; Wyatt, J.C. Automation bias: Empirical results assessing influencing factors. *Int. J. Med. Inform.* **2014**, *83*, 368–375. [CrossRef]
321. Cawley, G.C.; Talbot, N.L. On over-fitting in model selection and subsequent selection bias in performance evalu-ation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.
322. Steyerberg, E.W. Overfitting and optimism in prediction models. In *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*; Springer: Cham, Switzerland, 2019; pp. 95–112.
323. Navarro, C.L.A.; Damen, J.A.A.; Takada, T.; Nijman, S.W.J.; Dhiman, P.; Ma, J.; Collins, G.S.; Bajpai, R.; Riley, R.D.; Moons, K.G.M.; et al. Risk of bias in studies on prediction models developed using supervised machine learning techniques: Systematic review. *BMJ* **2021**, *375*, 2281. [CrossRef]
324. Tarafdar, M.; Vaidya, S.D. Challenges in the adoption of E-Commerce technologies in India: The role of organizational factors. *Int. J. Inf. Manag.* **2006**, *26*, 428–441. [CrossRef]
325. Shaikh, A.A.; Nasir Patel, A.; Shaikh, M.Z.; Chavan, C.R. SLRA: Challenges faced by SMEs in the adoption of E-commerce and sustainability in Industry 4.0. Acta Universitatis Bohemiae Meridionalis. *Acta Univ. Bohem. Meridionales* **2022**, *24*, 15–38. [CrossRef]
326. Zakaria, N.; Affendi, S.; Zakaria, N. Managing ICT in healthcare organization: Culture, challenges, and issues of technology adoption and implementation. In *Health Information Systems: Concepts, Methodologies, Tools, and Applications*; IGI Global: Hershey, PA, USA, 2010; pp. 1357–1372.
327. Koloski, D.; Porter, C.; Almand-Hunter, B.; Gatchell, S.; Logan, V. Data literacy in industry: High time to focus on operationalization through middle managers. *Harv. Data Sci. Rev.* **2025**, *7*. [CrossRef]
328. Jarrahi, M.H. Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. *Bus. Horizons* **2018**, *61*, 577–586. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Review

Learning-Based Viewport Prediction for 360-Degree Videos: A Review

Mahmoud Z. A. Wahba, Sara Baldoni * and Federica Battisti

Department of Information Engineering, University of Padova, 35131 Padua, Italy;
mahmoudza.wahba@phd.unipd.it (M.Z.A.W.); federica.battisti@unipd.it (F.B.)
* Correspondence: sara.baldoni@unipd.it

Abstract: Nowadays, virtual reality is experiencing widespread adoption, and its popularity is expected to grow in the next few decades. A relevant portion of virtual reality content is represented by 360-degree videos, which allow users to be surrounded by the video content and to explore it without limitations. However, 360-degree videos are extremely demanding in terms of storage and streaming requirements. At the same time, users are not able to enjoy the 360-degree content all at once due to the inherent limitations of the human visual system. For this reason, viewport prediction techniques have been proposed: they aim at forecasting where the user will look, thus allowing the transmission of the sole viewport content or the assignment of a different quality level for viewport and non-viewport regions. In this context, artificial intelligence plays a pivotal role in the development of high-performance viewport prediction solutions. In this work, we analyze the evolution of viewport prediction based on machine and deep learning techniques in the last decade, focusing on their classification based on the employed processing technique, as well as the input and output formats. Our review shows common gaps in the existing approaches, thus paving the way for future research. An increase in viewport prediction accuracy and reliability will foster the diffusion of virtual reality content in real-life scenarios.

Keywords: viewport prediction; 360-degree videos; streaming optimization

1. Introduction

In recent years, virtual reality (VR) technology diffusion has increased, offering highly immersive experiences that bring viewers into virtual environments. Unlike traditional 2D screens, VR allows users to interact with content, direct their focus, move their heads freely, and physically navigate within the virtual space. Therefore, users are provided with three or six degrees of freedom (DoF), plus the ability to interact with the environment. This enhances the sense of presence and gives the impression of physically being in the virtual scene. Virtual Reality (VR) experiences are typically accessed through a head-mounted display (HMD), with various options available, such as the Meta Quest, HTC VIVE, and Apple Vision Pro. The VR market has witnessed substantial growth, with projections indicating an increase from USD 47 billion in 2024 to an estimated USD 125.5 billion by 2030 [1].

In particular, 360-degree video has emerged as a popular form of content for VR experiences, also thanks to the diffusion of lightweight and low-cost capture devices. With 360-degree videos, viewers are surrounded by the media content, as if they were at the center of a sphere, thus allowing free exploration in all directions. This provides viewers with three DoF, allowing them to move their heads along three angles of movement, namely yaw, pitch, and roll, as shown in Figure 1.

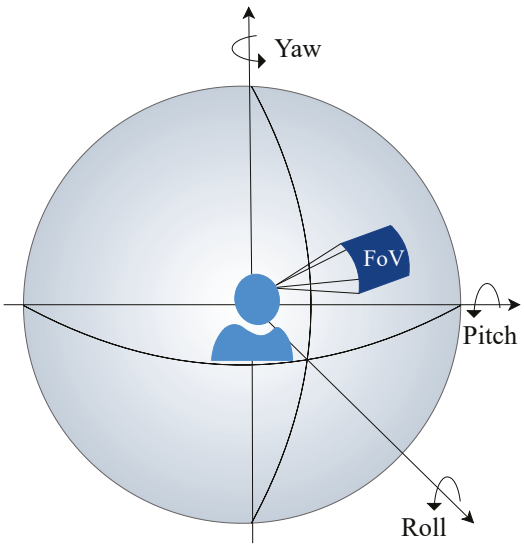


Figure 1. Diagram of 360-degree video content and potential angles of movement: yaw, pitch, and roll.

Although 360-degree videos enhance immersivity, the bandwidth requirements increase significantly with respect to traditional videos. For example, while streaming a 4 K video typically requires 25 Mb/s, delivering a 4 K resolution to each eye, allowing full 360-degree viewing, requires about 400 Mb/s [2]. Major streaming platforms like Facebook and YouTube usually transmit the entire 360-degree video frame [3]. However, the limitations of the human visual system, as well as the HMD's field of view (FoV), mean that users can only see a portion of the complete 360-degree video at any given moment, known as the viewport. This portion typically corresponds to less than 20% of the entire 360-degree video frame [4]. Therefore, sending the full frame is inefficient in terms of bandwidth usage, as it includes a significant amount of unused data. Moreover, transmitting a large amount of data can introduce transmission delays, leading to potential discomfort, such as cybersickness [5,6]. To address these issues, adaptive streaming technologies, such as tile-based adaptive streaming, have been developed to optimize bandwidth usage by predicting the viewport on which the user will focus [7]. These techniques split the 360-degree frames into fixed rectangular regions called tiles and prioritize the streaming of the tiles within the predicted viewport, while sending at a low resolution, or omitting entirely, the other portions of the frame. This method optimizes bandwidth usage. However, determining which areas to stream in a high resolution requires viewport prediction techniques. In addition, the prediction accuracy is of the utmost importance as the misprediction of the current viewport may highly impact users' quality of experience (QoE) [8].

In this scenario, artificial intelligence (AI)-based solutions come into play. In the last few years, many methods based on both machine learning (ML) and deep learning (DL) have been proposed to successfully perform viewport prediction. A possible strategy to classify viewport prediction techniques consists in considering the type of processed information. The first techniques mainly analyzed the previous head trajectories of the user to forecast subsequent head positions [9–11]. An evolution of this approach involves the integration of cross-user fixations to enhance predictions [12].

Approaches that rely on past head positions exhibit high efficiency when the prediction horizon is short (e.g., from 30 milliseconds to 1 s). However, their accuracy diminishes when attempting to make longer predictions, as the relevance of previous positions decreases over time. As an example, in [10], the prediction accuracy was around 96.6% when predicting 1 s ahead. However, when the prediction interval was extended to 2 s, the accuracy dropped to approximately 71.2%.

An alternative approach consists in exploiting saliency information. Saliency estimation allows the prediction of areas on which users will focus their attention [13]. Saliency algorithms usually take as input a video or a single frame and provide as output a map indicating the salient regions. Therefore, viewport prediction methods based on saliency select the next viewport as the one centered on the point with the highest value within the saliency map [14]. However, viewport prediction using saliency-based approaches has some limitations, as the predicted salient points exhibit faster and larger displacements compared to real head movements, which are generally smoother [14]. This makes the saliency-based approach less effective as a standalone method.

To enhance the prediction accuracy by leveraging the benefits of both strategies, hybrid methodologies have been explored, integrating past head positions with saliency estimation techniques. In addition, other information types can be complemented with head movements and saliency maps, such as eye gaze information.

An alternative perspective in classifying viewport prediction techniques consists in analyzing the type of employed algorithm. The types of learning architectures are extremely varied: the first methods presented in the state of the art employed less complex techniques such as regression or clustering. Over the years, however, the complexity of the employed approaches has increased, with the aim of improving the prediction performance and increasing the prediction horizon. Therefore, more complex architectures, such as Long Short-Term Memory (LSTM) and Transformers, have been proposed.

In addition, the output types are varied and may include the next head position, the next gaze position, the probabilities of tiles, or tiles within the viewport, as well as the quality level associated with each tile.

The choice of inputs and outputs, as well as of the specific algorithm, depends on several factors, such as the target complexity and the availability of data. In this survey, we propose a thorough review of the available techniques. Specifically, we classify existing state-of-the-art methods into three main categories, namely head movement-based, saliency-based, and hybrid approaches. The definition of the categories was established following a thorough investigation of the state-of-the-art methods and considering the types of information utilized by each approach for viewport prediction. Head movement-based methods rely only on users' past head movements to predict the future viewport. Saliency-based methods utilize users' visual attention, in the form of a saliency map, to perform the prediction. Hybrid approaches combine multiple sources of information—typically both head movement and saliency—to improve the prediction accuracy. Within each category, we identify a set of sub-classes based on the learning technique employed in each approach (e.g., regression, clustering, and Transformer models).

While prior surveys have broadly covered 360-degree video streaming and representation, this work is the first to systematically review viewport prediction methods as a standalone topic. We conduct a comprehensive review of the most notable and diverse viewport prediction methods from the past 10 years.

We selected papers based on the following inclusion criteria:

1. Recency: We focused on papers published within the last 10 years (2016–2025) in order to reflect the development of viewport prediction, starting from regression methods, which were already being used in 2016, and ending with modern approaches employed in the last year.
2. Diversity of approaches: We intentionally included works using a variety of strategies, namely head movement-based, saliency-based, and hybrid approaches, to ensure broad coverage of the field and show the strengths and drawbacks of every approach.
3. Technique variety: For each category, we selected works that employed a range of techniques, such as regression, clustering, LSTM, Transformer models, reinforcement

learning (RL), and graph neural networks (GNNs), to reflect the technical diversity in the state-of-the-art methods.

To perform paper selection, we applied the following exclusion criteria:

- Duplicates (across databases or different papers using the same viewport prediction approach);
- Papers that were not written in English;
- Papers with no citations;
- Papers not addressing omnidirectional content;
- Papers focusing on proposing new streaming and/or bit allocation techniques employing existing viewport prediction approaches.

We used the Scopus and IEEE Xplore databases as these cover the majority of high-impact computer science and multimedia venues relevant to viewport prediction. We used the following search keywords: "viewport prediction" AND ("360 video" OR "360° video" OR "360-degree video" OR "omnidirectional video"). Boolean operators AND/OR were applied consistently across both databases. Paper selection was performed independently by at least two reviewers, with disagreements resolved through discussion to reduce bias. We provide the PRISMA flow diagram in Figure 2.

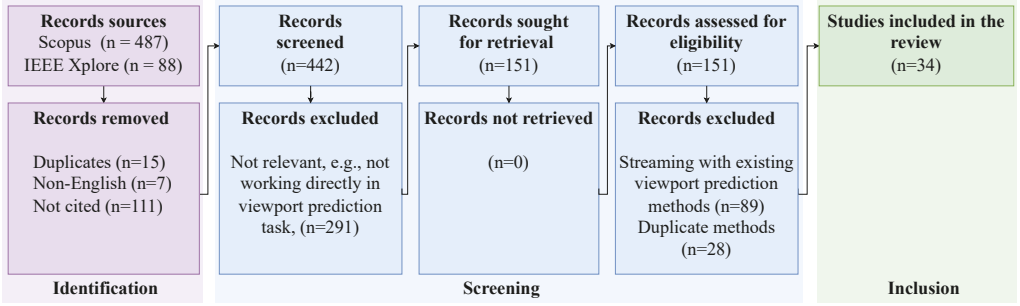


Figure 2. PRISMA flow diagram of the performed search.

The remainder of the paper is organized as follows. In Section 2.1, we review the viewport prediction techniques that process only head movement data; in Section 2.2, we describe the methods based on saliency information, and, in Section 2.3, we report the hybrid approaches. Later, in Section 3, we provide an overview of existing datasets for 360-degree videos. In Section 4, we present a comprehensive analysis of the evaluation metrics adopted by state-of-the-art methods and outline the baseline or comparison methods used by each approach. In addition, we provide a discussion of current trends and gaps in Section 5, together with some recommendations for future research directions. Finally, Section 6 concludes the paper.

2. Viewport Prediction

In this section, we review the approaches presented in the state of the art by performing broad classification into (i) head position approaches, (ii) saliency-based approaches, and (iii) hybrid approaches. For each of these categories, we provide differentiation based on the employed processing algorithm and the type of output. A general overview is provided in Figure 3.

The left side of the figure shows the possible input types. As a first option, the input might represent the traces of eye movements regarding the 360° video (eye gaze scanpaths). Alternatively, head-related information could be employed in two forms. In the first option, head movements are represented as points on the sphere or on the 360-degree equirectangular projections (ERPs) of the frames, thus realizing what is known as a head scanpath. In the second option, the input consists of head orientation traces, where orientation angles

are expressed with respect to a head-centered system. Users' viewing behavior can also be represented as viewport maps, i.e., heatmaps in the same format as the video frame that highlight the viewport region. Another option is to provide the 360-degree video frames as input, which typically occurs for saliency-based methods. Finally, points of interest (PoIs), i.e., regions of the 360° frame identified as relevant, could be extracted in advance and then provided as input to the viewport prediction algorithm.

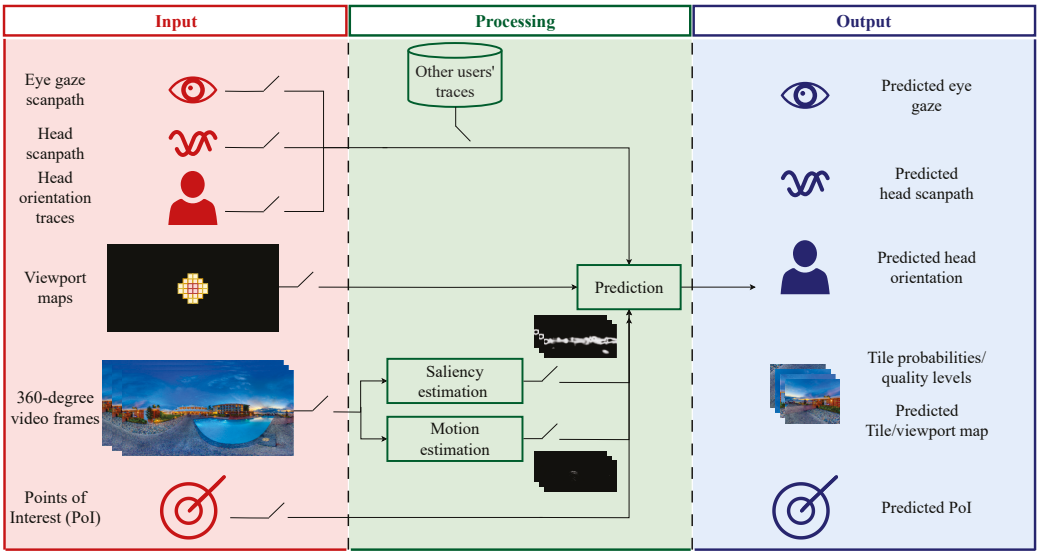


Figure 3. Overview of input/output combinations for viewport prediction.

The central column of Figure 3 shows the possible processing strategies. Viewport prediction could be realized by directly processing one or more raw inputs or by pre-processing the video frames for the extraction of saliency and/or motion information. In addition, some methods rely only on data from a single user, while others integrate the data of multiple viewers. Additionally, some approaches combine different modalities.

Finally, the outputs depend on both the input type and the processing method. Possible outputs include the predicted eye gaze, head scanpaths, or orientations. Additionally, algorithms could provide as output, for each frame tile, either its probability of being part of the viewport or its assigned quality level. In other cases, viewport prediction methods could directly output the future PoIs or a set of predicted tiles or viewports.

A summary of the reviewed approaches, together with the corresponding classification, is provided in Table 1.

2.1. Head Position Approaches

The core principle of viewport prediction based on head movements is to record historical data of head displacements to predict the following ones. There are two main sub-categories of head-based viewport prediction: (i) single-user and (ii) multi-user approaches. The former performs movement prediction for a user U_i based on the previous movements of the same user. The second class works under the assumption that users will behave similarly when watching the same video, so that the movement trajectories of different users, $U_j \neq U_i$, are used to predict the head trajectory of the i-th user. In the second case, the users employed in the training process are usually grouped into sets so that the set that is most similar to the i-th user is employed for prediction. In the following, we give a detailed discussion of the different processing approaches that employ the head movement strategy for viewport prediction.

2.1.1. Regression Approaches

Regression approaches represent some of the first ML-based techniques applied to viewport prediction through head movement analysis. Among the first contributors, Qian et al. proposed a sliding window framework [10]. Specifically, they defined a window size of 1 s for the analysis of head movements, in terms of head orientation, with the aim of predicting the following 2 s of movement data. To this end, they applied three methods: (i) average, (ii) linear regression (LR), and (iii) weight linear regression (WLR). The former is a simplistic model where the average head position is computed in the analysis time window and used as a prediction for the next 2 s. The second approach consists in training an Linear Regression (LR) model on the analysis window by weighting all training samples equally. The third approach refines the LR technique by assigning a larger weight to more recent samples. The authors compared the three techniques by performing a subjective experiment with a custom dataset consisting of four short 360-degree videos sourced from YouTube and watched by five users. The experiments were conducted using a Google Cardboard viewer with a mobile phone device. Accuracy was used as an evaluation metric. The results revealed successful predictions for short-term intervals (from 0.5 to 1 s), with Weighted Linear Regression (WLR) achieving accuracy of 96.6% $\pm$ 2.0% for 0.5 s and 92.4% $\pm$ 3.7% for 1 s. However, prediction for longer intervals (i.e., 2 s) proved to be more challenging, as the WLR method achieved accuracy of 71.2% $\pm$ 7.6%. As expected, WLR demonstrated higher accuracy compared to the other methods, with LR exhibiting better accuracy than the average method. Additionally, the authors measured the bandwidth savings in cases where viewport prediction can be successfully performed, i.e., testing the best-case scenario where the next viewport is known. Their analysis revealed that viewport prediction allows a bandwidth reduction of 80%, thus clearly showing its impact [10].

This study was extended in [11] by collecting a new custom dataset containing head traces from 130 participants who watched 10 360-degree videos. The acquisitions were performed using a Samsung Gear VR headset with a Samsung Galaxy S8 device. In this case, the authors used an analysis window of 1.5 s and aimed at predicting the head movements of the next 3 s. To evaluate the model, they used the same accuracy metric as applied in the previous study [10]. Four models were investigated: (i) static, which assigns to the predicted head position the current location, thus assuming that no movement is performed by the user; (ii) LR; (iii) ridge regression (RR), a more complex version of LR that adds a regularization term to reduce overfitting; and (iv) a support vector regressor (SVR), which utilized support vector machines (SVMs) for regression. The static approach will be referred to as naive in the following. Based on the achieved performance, the authors selected LR to perform a one-second prediction and Ridge Regression (RR) to extend the prediction up to the third second.

As an additional contribution, the authors proposed Flare, a practical streaming system for 360-degree videos on mobile devices incorporating viewport prediction. Flare has been evaluated in terms of the stall duration, average bitrate, and average quality level. Based on these parameters, Flare enhanced the quality by up to 18 times on WiFi compared to the no-prediction approach. Additionally, it achieved bandwidth reductions of up to 35% and quality enhancements of up to 4.9 times on LTE.

Table 1. Viewport prediction approaches.

Category		Reference	Input	Output	Technique	Horizon
Head movement approach (Section 2.1)	Regression (Section 2.1.1)	Qian et al. 2016 [10]	Head orientation traces	Head orientation	Average LR, WLR	2 s
		Bao et al. 2016 [9]	Head orientation traces	Head orientation	Naive, LR shallow NN	1 s
		Qian et al. 2018 [11]	Head orientation traces	Head orientation	Naive, LR RR, SVR	3 s
		Ban et al. 2018 [12]	Head orientation traces (current and previous users)	Tile probabilities	LR, KNN	6 s
	Clustering (Section 2.1.2)	Petrangeli et al. 2018 [15]	Head orientation traces (current and previous users)	Head orientation	Spectral clustering	10 s
		Nasrabadi et al. 2020 [16]	Head orientation traces (current and previous users)	Head orientation	Naive, custom clustering	10 s
		Dziubinski et al. 2024 [8]	Video frames, tiles in viewports (previous users)	Tile quality level	KNN, DBSCAN	1, 2 s
	LSTM (Section 2.1.3)	Li et al. 2023 [17]	Head orientation traces	Head orientation	BiLSTM, metalearning	5 s
		Mahmoud et al. 2024 [18]	Head orientation traces	Tile probabilities Tile quality levels	LSTM	1 s
		Feng et al. 2025 [19]	Head scanpath	Tile probabilities	LSTM	1–6 s
	Transformer (Section 2.1.4)	Chao et al. 2021 [20]	Head scanpath	Head scanpath	Transformer	5 s
		Wang et al. 2024 [21]	Head scanpath	Head scanpath	Transformer DRL	1–5 s
Saliency approach (Section 2.2)	Saliency estimation	Aladagli et al. 2017 [14]	Video frames	Saliency map	GBVS	2 s
	Transformer	Alice et al. 2024 [22]	Video frames	Tile probabilities	TranSalNet VST	1 s
Head movement with saliency approach (Section 2.3)	LSTM (Section 2.3.1)	Fan et al. 2017 [23]	Head orientation traces, motion and saliency maps	Tile probabilities	LSTM, CNN Optical flow	1 s
		Nguyen et al. 2018 [24]	Viewport maps, saliency maps	Viewport map	LSTM PanoSalNet	2.5 s
		Li et al. 2018 [25]	Head orientation traces, navigation speed, video frames	Tile probabilities	LSTM, FSM, optical flow	1 s, 2 s
		Xu et al. 2018 [26]	Eye gaze scanpath, video frames, current and past FoV images	Gaze displacement	LSTM, FlowNet2, SalNet	1 s
		Fan et al. 2020 [27]	Head orientation traces, motion and saliency maps	Tile probabilities	LSTM, CNN, Optical flow	1 s
		Feng et al. 2020 [28]	Head orientation traces, video frames	Tiles in viewport	LSTM, CNN	2 s
		Rondón et al. 2022 [29]	Head orientation traces, saliency maps	Head orientation	LSTM PanoSalNet	5 s
		Nguyen et al. 2023 [3]	Viewport maps, saliency maps	Viewport map	LSTM PanoSalNet	1 s
		Manfredi et al. 2023 [30]	PoIs	PoI	LSTM	3 s

Table 1. *Cont.*

Category		Reference	Input	Output	Technique	Horizon
Head movement with saliency approach (Section 2.3)	Conv-LSTM (Section 2.3.2)	Wang et al. 2022 [31]	Viewport maps, video frames	Viewport map	ConvLSTM, CNN	2 s
		Dong et al. 2023 [32]	Eye gaze heatmaps (current and previous users)	Eye gaze heatmaps	Conv-LSTM SE-Unet	10 s
		Wang et al. 2024 [33]	Video frames Viewport maps (current and previous users)	Viewport map	ConvLSTM, CNN, Clustering	-
		Zhang et al. 2025 [34]	Eye gaze scanpath, video frames	Tile probabilities	ConvLSTM, GCN	-
	GRU (Section 2.3.3)	Li et al. 2023 [35]	Video frames, viewport maps (previous users)	Viewport map	S-SPCNN, T-SPCNN, SP-ConvGRU	2 s
		Setayesh et al. 2024 [36]	Head orientation traces, video frames	Viewport tiles	PAVER, GRU	-
	Transformers (Section 2.3.4)	Zhang et al. 2022 [37]	FoV images and head and eye gaze scanpaths	Head scanpath	Transformer, CNN	10 s
		Gao et al. 2024 [38]	Video frames, head scanpath	Head scanpath	PAVER, LSTM, Transformer	1, 2–5 s
		Guo et al. 2024 [39]	Video frames	Head scanpath	Bidirectional Transformer, residual network	-
	GNN (Section 2.3.5)	Xu et al. 2023 [40]	Video frames, head orientation traces (current and previous users)	Tile probabilities	GNN, LSTM, SalNet, K-means, FlowNet	3 s
	RL (Section 2.3.6)	Xu et al. 2019 [41]	Video frames, head scanpaths (current and previous users),	Head scanpath	DRL	0.03 s

Regression for viewport prediction has been applied also in [9]. Specifically, the authors compare three approaches. The first is the naive technique. The second model is LR, and the third algorithm is a shallow neural network model consisting of three layers and 50 hidden neurons. In this study, the next viewport is predicted by estimating the pitch and yaw angles. As users tend to explore content horizontally (yaw change) rather than vertically (pitch change), the focus is placed on yaw prediction, while the pitch is omitted. To evaluate the prediction accuracy, four metrics have been employed: the mean error, root mean square error (RMSE), 99th percentile, and 99.9th percentile. The shallow neural network model demonstrated superior performance compared to the baseline approach, yielding approximately a 50% reduction for all metrics for a 0.2 s prediction window. With respect to the LR method, the improvement was smaller, corresponding to a 0.09 and 0.32 decrease in the mean error and Root Mean Squared Error (RMSE) and a reduction of 1.64 and 3.25 in the percentiles. The models were tested using a custom dataset generated by 153 viewers. Specifically, 35 subjects watched 16 video clips, and 118 watched three to five randomly selected video clips using an Oculus DK2 headset. The prediction results were then employed to select the area to transmit, thus controlling the amount of content delivered to the viewer. Specifically, if the prediction accuracy is high, only the content related to the viewport is sent to the user, whereas, if the accuracy is low, more content is sent to maintain good quality. In the extreme scenario of poor accuracy, such as a high failure ratio, the entire frame might be sent. The authors performed viewport prediction for 1 s windows, demonstrating good performance within $100 - 500$ milliseconds. With a prediction window of 0.2 s, they achieved a 45% reduction in bandwidth consumption within a 1% failure ratio [9].

An alternative approach for regression has been presented in [12]. Specifically, the authors propose CUB360, which improves prediction for longer periods by complementing the personalized analysis of user behavior with information from multiple users. In more detail, LR is employed to forecast the user's head orientations in future frames using the target user's past trajectory data. Then, the prediction is refined by integrating head

orientations from other users who have already viewed the same video, through K-Nearest Neighbors (KNN), by selecting the K closest traces. Finally, a probability voting mechanism is applied, combining the results from LR and KNN. Model evaluation has been performed using a dataset consisting of 18 videos viewed by 48 participants [42]. They tested various sizes of prediction windows, ranging from 2 to 6 s, analyzing head orientations within a sliding window of 1 s. To evaluate the proposed method, four evaluation metrics have been used: (i) the viewport peak signal-to-noise ratio (PSNR), where the Peak Signal-to-Noise Ratio (PSNR) is computed only over the user's viewport, using the ground truth as a reference; (ii) viewport deviation (v_D), which calculates the percentage of blank area, i.e., empty tiles, inside the viewport, so that the accuracy is computed as $1 - v_D$; (iii) viewport quality variance, which utilizes the coefficient of variation to evaluate the consistency of visual quality across the viewport within a single video segment—the coefficient of variation is defined as $\text{CV} = \left(\frac{\sigma}{\mu}\right) \times 100\%$, where σ indicates the standard deviation and μ the mean; (iv) bandwidth occupation, which calculates the total bitrate consumed per segment, with higher values indicating increased data utilization. The proposed method was compared to the standalone LR approach, without incorporating KNN. Their findings reveal significant improvements compared to the LR approach, with an absolute improvement (i.e., $accuracy(LR) - accuracy(CUB360)$) of 20.2% and a relative increase (i.e., $(accuracy(LR)/accuracy(CUB360)) \times 100$) of 48.1%. Additionally, CUB360 enhanced the viewport PSNR by 30.28% with respect to LR and improved bandwidth usage by 32.26% [12]. This work demonstrates that clustering approaches represent a valuable technique for viewport prediction, as will be detailed in the following section.

2.1.2. Clustering Approaches

Compared to regression-based studies, clustering approaches have been investigated to forecast the viewport over longer durations, extending up to 10 s. Based on the assumption that many users behave similarly when watching 360-degree videos, the behavior of viewers who watch the same videos can be grouped using clustering techniques. Subsequently, the viewport can be predicted for a new viewer by identifying the cluster to which his/her behavior is aligned [15,16].

For instance, in [15], users' head orientations over time have been represented as trajectories, and spectral clustering has been applied to group similar trajectories for the same 360-degree video. From each cluster, a trend trajectory is selected as a representative of the set, showing the average behavior of trajectories within that cluster. These trend trajectories are used to predict the future head orientations of new users watching the same 360-degree video. The prediction was performed in intervals of 1, 5, and 10 s. The dataset employed in the study was the one presented in [9]. To evaluate the proposed method, two metrics were used. The first metric was the great circle distance, also known as the orthodromic distance (i.e., the shortest distance between two points on the surface of a sphere), between the center of the predicted viewport and the ground-truth viewport. The second metric measured the ratio of the overlapping area between the predicted viewport and the ground-truth viewport. The authors compared the proposed model with other state-of-the-art approaches, including the LR model introduced by Bao et al. [43], CUB360 [12], and the naive baseline approach. Surprisingly, the comparison highlighted that, for 1 s predictions, the clustering approach and CUB360 were outperformed by the naive technique. However, when the prediction duration was larger, the performance of the clustering-based approach increased. For example, for 5 s predictions, the clustering approach successfully predicted the viewport area (in terms of 50% overlap between the actual and predicted viewport) for 70% of viewers, whereas the naive approach predicted the same overlap percentage for only 50% of viewers [15].

The approach presented in [15] described the position of the viewport using the roll, pitch, and yaw angles and tracked how each angle changed over time as an independent trajectory. In [16], a different clustering strategy has been employed. Specifically, the viewport has been represented as a circle, mapped to the 360-degree scene, and represented through quaternions, whose diameter represents the FoV. Viewers who have watched the same video are grouped together, using orientation traces of 1 s, if the distance between their viewport centers is less than 30 degrees for at least 90% of the time window. Every second, a new set of clusters is created. To predict the next viewport for the current user, two operations are performed: (i) cluster assignment and (ii) cluster-based viewport prediction. First, a user is assigned to a cluster if his/her current viewport center (i.e, at time t) is within 30 degrees of one of the cluster's viewport centers. Then, the viewports at $t + 1$ for the users of the cluster are extracted. The quaternions associated with the centers of the viewports are averaged using the method proposed in [44] and used as a prediction of the center of the user's predicted viewport. In the event that a user cannot be assigned to a cluster, the predicted viewport is set to be equal to the current one. In [16], prediction has been performed for 1, 3, 5, and 10 s using a dataset of 28 videos, each watched by 30 participants [45]. To evaluate the proposed approach, the effectiveness of prediction was measured by computing the overlap between the predicted viewport and the actual one, i.e., the Field of View (FoV) overlap. The model has been compared with a custom LR approach along with the naive approach. The authors show that their algorithm outperforms both methods in more than 40% of the dataset when using a prediction window of 5 s or more. However, for prediction periods of less than 3 s, setting the last viewport as the current prediction yields better results than their proposed method. Additionally, when analyzing the videos in the dataset, they discovered a strong correlation between the prediction accuracy and video content. Specifically, the clustering-based method performs better when a region of interest (RoI) is present in the video, e.g., a moving object, as it captures users' attention and maintains their focus.

Clustering has been employed for viewport prediction in [8]. As viewport misprediction strongly impacts the user experience, the authors propose to enlarge the predicted viewport by including the surrounding region in the high-quality set of tiles, thus realizing viewport extension. Starting from a global repository stored on the streaming server, the authors propose extracting a local repository by identifying the subset of users that are more similar to the current one through KNN and DBSCAN clustering. Both repositories are employed to compute a score for each tile to rate its popularity. The latter is employed to assign a quality factor to each tile. Specifically, an extension region of tiles is defined to assign high quality to the tiles surrounding the predicted viewport. The method's performance has been tested using the dataset presented in [46], consisting of 10 videos watched by 30 users. The prediction horizon was set to 1 and 2 s, and the bitrate assigned to the actual viewport was used as an evaluation metric. The experimental results show that KNN has superior performance with respect to DBSCAN. In addition, the best-performing repository, i.e., global vs. local, was dependent on the video content. For slow-paced and content-rich videos, the local repository led to better performance, as global tiles might be confused due to the high number of points of interest. In contrast, the two repositories performed similarly for fast-paced and centralized content, such as rollercoaster videos. Finally, the global repository performed better on small tiles, while the local one achieved superior performance on large tiles.

2.1.3. LSTM

Clustering methods are useful in predicting the behavior of new users by comparing them to previous users who have watched the same videos. However, these techniques

have two limitations: (i) viewing trajectories of other users might not be available and (ii) the underlying assumption of similarity between users' viewing behaviors might not hold. To address these issues, the use of LSTM methods has been explored. LSTM is a type of neural network designed to learn from sequences, thus allowing the analysis of head movements or gaze over time. This capability enables more accurate predictions of future viewports. Furthermore, LSTM can make individual predictions for each user, rather than relying on group data, which adds flexibility and personalization to the predictions.

Due to these properties, an LSTM model was selected for viewport prediction in [18]. The proposed method takes as input the pitch and yaw angle traces and produces as output the tile probabilities, together with the corresponding quality levels. Specifically, the LSTM is employed to predict tile probabilities from past user data, and a greedy method is applied to select the corresponding quality levels. All tiles are initialized with the lowest quality level, which is incrementally improved based on the resulting user's perceived quality. The quality is opportunistically updated by taking into account the corresponding bitrate increase. Performance evaluation has been conducted with the dataset presented in [47], consisting of 60 videos watched by 221 users, with a prediction horizon of 1 s. To evaluate the proposed method, the viewport-weighted PSNR (VWS-PSNR) metric was utilized. The Viewport Weighted PSNR (VWS-PSNR) is a user-centric quality metric that assesses video quality specifically within the user's viewport, rather than across the entire 360° frame. It employs a weighted mean squared error with spherical correction calculated only on the viewport pixels to better reflect the perceived quality. The authors simulated a multi-user 360° video streaming system using 20 360-degree video sequences, each segmented into 1 s intervals, along with head movement traces from 20 users. To validate the effectiveness of the proposed greedy method for quality selection, they compared it with the methods described in [48–50]. The results demonstrated that the proposed method significantly outperformed the baselines in terms of viewport quality, especially under bandwidth-constrained conditions.

An LSTM-based approach is leveraged also in [19], where the authors define *Stoc-360*, a multi-window probabilistic viewport prediction and tile-level bitrate adaptation streaming algorithm. Instead of relying on a binary decision to determine whether a tile will appear in the next viewport, Stoc-360 assigns each tile a probability of being viewed, where a higher probability indicates a higher likelihood of being in the upcoming viewport. Recognizing that prediction becomes more difficult as the prediction window increases, the authors use multiple prediction models, each tailored to a specific prediction window (e.g., 1 s, 2 s, and so on). To further mitigate prediction challenges with longer windows, they introduce a patience pattern, which temporarily pauses video downloading to reduce the prediction window length. Their viewport prediction model is based on a three-layer LSTM with 128 hidden units that takes as input the head trajectory on the sphere, also referred to as the head scanpath. To evaluate the QoE, they consider four metrics: viewport quality (higher bitrate yields better visual quality), spatial variation (bitrate uniformity across tiles in the viewport), temporal variation (bitrate consistency over time), and re-buffering time (playback interruptions). Experiments were conducted on the dataset presented in [51], consisting of 19 videos watched by 57 users. Stoc-360 has been compared and evaluated against three state-of-the-art models: Flare [11], TBAS [52], and DRL360 [53]. Stoc-360 outperformed all baselines, achieving a 16.75% higher Quality of Experience (QoE) score than TBAS, a 21.46% gain over Flare, and a 48.80% increase relative to DRL360.

As described in the previous paragraphs, LSTM models are adept in learning patterns from sequences, such as how a user's gaze moves over time in 360-degree videos. However, standard LSTM only looks at the past to make predictions. To successfully perform viewport prediction, it may also be helpful to look at future information in the sequence. This

is where Bidirectional Long Short-Term Memory (BiLSTM) comes into play. A BiLSTM network processes the sequence in both the forward and backward directions, so that it can learn from the full context.

A BiLSTM model has been applied to viewport prediction in [17], where it has been integrated with metalearning. Metadata are incorporated to tailor personalized prediction models to individual users' preferences. To better understand user preferences, user characteristics and behavior when viewing 360-degree videos have been analyzed using the algorithm presented in [54]. The prediction horizon for viewport prediction was set to 5 s. The experimental evaluation encompassed five datasets, with the largest comprising 208 videos watched by 30 users [26], along with four additional datasets: one with 76 videos viewed by 58 users [41], another with 11 videos viewed by 48 users [24], a third with 10 videos watched by 25 users [23], and the last corresponding to the dataset presented in [46]. To evaluate the proposed method, several metrics were used. In more detail, the authors employed both standard indicators such as accuracy, the F1-score, and the Intersection over Union (IoU) (whose interpretation in viewport prediction is provided in Section 4.1) and task-specific metrics. Specifically, they considered the mean FoV overlap, which is here defined as the area of intersection between the predicted and ground-truth tiles divided by the area of union of predicted and ground-truth tiles; the intersection angle error, which measures the distance between the predicted point and the actual point on a sphere; and the orthodromic distance. Li et al. compared their model with several state-of-the-art methods, including those presented in [23–26,29,41,55]. They reported that their model achieves better performance by enhancing the orthodromic distance by 10% to 50% in a prediction window of 5 s. The authors highlighted that the performance increase was higher in datasets with intense user head movements.

2.1.4. Transformers

Unlike LSTM models, which process sequences step by step and rely on memory to retain important information, Transformers use a self-attention mechanism that allows them to consider the entire sequence at once. This makes them faster and more effective in capturing long-term dependencies, especially in tasks involving long sequences [56,57]. For this reason, Transformers have achieved great success in many fields and are now widely used in various applications, including viewport prediction. For instance, in [20], the Viewport Prediction Transformer (VPT360) model has been introduced. It forecasts the future positions of viewport centers by analyzing the user's previous viewport positions over time and representing them as head scanpaths. The model examines the evolution of the scanpath and captures the dependencies between positions. In [20], a scanpath of 1 s has been employed to estimate the future viewport positions for 5 s. VPT360 was evaluated using the datasets presented in [41,42,51], and its performance was assessed using three metrics: the average great circle distance; the average ratio of overlapping tiles, which measures the percentage of overlap between the predicted and ground-truth tiles; and the mean overlap, which computes the average intersection over union between the predicted and ground-truth areas for a given prediction window. The VPT360 model was compared with [29] on the dataset presented in [51] and demonstrated superior performance for the entire 5 s window. Moreover, VPT360 was compared with other models, such as [16,24], as well as [29], on the dataset presented in [42], outperforming these methods for prediction horizons larger than 0.5 s in terms of the average ratio of overlapping tiles.

Another Transformer-based approach was presented in [21]. The authors propose exploiting attention mechanisms to extract spatiotemporal information and predict multiple candidate head trajectories based on historical head data. Then, deep reinforcement learning (DRL) is employed to maximize a QoE objective based on the network conditions.

Specifically, three candidate viewports are predicted, and each is assigned a different bitrate by the Deep Reinforcement Learning (DRL) module. The latter operates based on a QoE objective composed of four components: (i) viewport quality, measured as the bitrate assigned to the viewport region; (ii) viewport temporal variation, measured as the bitrate difference of the viewport region in consecutive instants; (iii) viewport spatial variation, measured as the quality difference between the different viewport regions; and (iv) a re-buffering measure. The proposed method has been evaluated using the dataset presented in [51] with 1 s data as input and a prediction horizon between 1 and 5 s. To evaluate viewport prediction, two metrics were used: the great circle distance and the average great circle distance. The proposed method was compared with many approaches, including [20,29]. It achieved the smallest average error across all time windows.

2.2. Saliency-Based Approaches

The second category of input for viewport prediction algorithms is represented by saliency information.

As an example, in [14], a saliency-only viewport prediction approach has been presented. Specifically, the Graph-Based Visual Saliency (GBVS) algorithm has been employed to predict the saliency of each viewport during the viewing session, under the hypothesis that the most salient area in each frame represents the next viewport. Therefore, for each frame, a single center of attention was selected using the technique proposed in [58], where the location of attention is designated by the pixel with the highest saliency value. The evaluation was performed based on the dataset presented in [9]. Cross-correlation analysis was performed between the predicted attention center and real head movements to assess their relations. However, the results indicate that this approach is not robust, as the predicted attention points exhibit overly rapid and extended movements compared to real head movements, which are smoother.

Viewport prediction based solely on saliency information has been explored also in [22]. Viewport prediction was performed using a pre-trained 2D Transformer-based saliency model, applied directly to 360-degree video frames without any additional training or fine-tuning specific to 360-degree content. Specifically, the 360-degree frames in ERP format were processed through a 2D saliency estimation model, after which each frame was divided into 9×16 rectangular tiles. The average saliency value within each tile was calculated and employed as a measure of the probability that the tile would be viewed by a user. For saliency estimation, two 2D models were employed: TranSalNet [59], which combines Transformer layers with convolutional layers, and the Visual Saliency Transformer (VST) [60], which relies solely on a Transformer architecture without additional convolutional components. The authors introduced an algorithm that computes the weighted sum of the saliency values of each tile along with the saliency values of its neighboring tiles to determine the final bitrate for each tile. The experimental evaluation was conducted using the publicly available dataset proposed in [42], and the model proposed in [24] was used as a baseline for comparison. The methods were compared in a live streaming scenario, and three metrics were employed to evaluate the QoE: the average bitrate in a user's viewport for each frame, the variation in the bitrate within the viewport for each frame, and the variation in the bitrate across successive tiles. The proposed method is suitable for real-time live streaming, as it does not require previous head movement data; it enables real-time bitrate allocation and achieves promising performance. Moreover, the authors demonstrated that their method can achieve up to a 50% increase in bitrate for the predicted viewport compared to standard 360-degree viewing. However, with respect to [24], which benefits from previous head orientation information, the obtained results are less accurate.

Indeed, methods based only on saliency do not account for the physical limitations of head movements, which are inherently learned by approaches based on head trajectories. In addition, as 360-degree videos are explored through HMDs, head tracking can be performed by extracting telemetry information from the headset, without placing an additional burden on the video experience. Therefore, instead of relying solely on saliency, head movements and saliency analyses can be integrated to obtain a hybrid viewport prediction approach, as will be detailed in the following section.

2.3. Hybrid Approaches

As previously discussed, various ML and DL approaches have been explored to predict the future viewport. For instance, some methods rely solely on the past head movements of the viewer to predict the upcoming one, but these techniques are most effective for short-term predictions [9–11]. Another strategy involves clustering algorithms, which consider the previous head movements of the target user alongside those of other viewers who have watched the same video, enabling predictions up to 10 s. However, these methods are frequently considered unsatisfactory and impractical, as they rely on the availability of a set of previous viewing trajectories for a target video. Therefore, this type of approach may not be applicable when dealing with a new 360-degree video. Furthermore, the efficacy of this strategy depends on the type of content [15,16]. An alternative approach consists in exploiting the saliency map generated from the video frames to anticipate the future viewport, focusing on the most salient regions. However, this approach often produces unsatisfactory outcomes, particularly in achieving smooth transitions in the predicted viewport [14]. Consequently, there has been a push to explore novel methods for viewport prediction, combining the previous head movements of the target user with saliency maps of the 360-degree video. Numerous Deep Learning (DL) techniques have been investigated for this purpose, as will be detailed in the following.

2.3.1. LSTM

Fan et al. were among the first to incorporate head orientations, saliency maps, and motion maps, i.e., optical flow information, into viewport prediction [23]. Their method used an LSTM network [61], which is well suited for learning dependencies among video frames. The study involved evaluating two types of networks: the orientation-based network and the tile-based network. The orientation-based network takes as input the concatenation of the previous user's head orientations and the downsampled version of the saliency map and motion map (64×64 pixels), whereas the tile-based network uses the viewed tiles instead of head orientations. Motion maps were generated using Lucas–Kanade optical flow [62] between consecutive frames, while saliency maps were estimated using a pre-trained CNN based on VGG-16 [63], as proposed in [64]. Due to the resource-intensive nature of motion and saliency map estimation, these procedures were performed offline. The prediction horizon was set to 1 s and the simulations were performed using a custom dataset of 10 360-degree videos viewed by 25 users through the Oculus Rift Head-Mounted Display (HMD). The proposed method was evaluated using many metrics, including accuracy. Due to the novelty of their approach at the time, the authors compared it against three baseline methods. The first was the naive approach. The second used saliency, selecting tiles whose average saliency values exceeded a certain threshold to represent the predicted viewport. The third was the Dead Reckoning (DR) method proposed in [65], where the future viewport is estimated based on the viewer's current head orientation and velocity. The comparison among the three approaches highlighted that the proposed one offered several benefits, such as an average bandwidth reduction between 22% and 36% and shorter buffering times, with reductions of about 43% [23].

The approach presented in [23] has been extended through the definition of the Future-Aware model [27]. Specifically, the authors included information about video content and addressed distortion effects in projection. To tackle the distortion caused by projection operations (e.g., equirectangular, cubic, and rhombic dodecahedrons [66,67]) and their impacts on saliency detection, the overlapping virtual viewport technique has been employed. This technique involves generating virtual viewports covering the entire sphere, each with a size of 60 degrees and a sampling angle of 30 degrees. Saliency estimation is performed on these viewports using methods developed for 2D images, and the resulting outputs are stitched together to create the final saliency map. The proposed model has been evaluated with a prediction horizon of 1 s using a custom dataset of 10 360-degree videos watched by 50 users collected with the Oculus Rift HMD. For performance evaluation, several metrics were used, including the missing ratio and the unseen ratio, which represents the ratio of tiles that arrived at the client but were not watched compared to the total number of transmitted tiles. In addition, the authors measured the bandwidth consumption and the peak bandwidth, which indicates the highest bandwidth consumption recorded during this process. Additionally, they considered the V-PSNR defined in [68], along with the total re-buffering time, which measures the overall time spent re-buffering during playback. For comparison, they used the naive approach and the DR method proposed in [65]. The authors proved that their approach outperforms other algorithms in terms of bandwidth efficiency and re-buffering time while achieving comparable video quality. The authors show that the proposed method eliminates re-buffering events when the available bandwidth exceeds 25 Mbps, which is approximately 10 Mbps lower than required by alternative algorithms [27].

LSTM networks have been employed also in [24] to predict future head movements, utilizing saliency maps and head orientations as inputs. To facilitate processing, they represented head orientation though a tile-based map in order to align it with the saliency map. For saliency estimation, they introduce the PanoSalNet model inspired by Deep Convnet [69], a convolutional neural network for saliency detection in 2D images. PanoSalNet comprises 8 convolutional layers and one deconvolutional layer, with the initial 3 layers having the same structure as VGG-16 [63]. The subsequent 5 layers are initialized and trained on the SALICON dataset [70], a saliency map dataset for 2D images. The prediction method proposed in [24] forecasts head orientations up to 2.5 s. The overall model pipeline has been trained and evaluated using a custom saliency dataset consisting of 11 videos viewed by 48 users. The proposed method was evaluated against various scenarios presented by the authors in terms of prediction accuracy. The first scenario used only the result from Deep ConvNet as the output of viewport prediction. The second scenario employed the result of PanoSalNet as the output of viewport prediction. The third scenario used Deep ConvNet for saliency estimation, in conjunction with the head orientation, to make predictions. The authors examined the impact of the prediction window on the accuracy. They found that their model performed differently across scenarios, with the accuracy dropping from around 80% with a prediction window of 0.5 s to about 45% with a prediction window of 2.5 s. Additionally, they studied how the content of the video influenced the viewport prediction accuracy. They reported that the best performance, with accuracy of 85%, was achieved with static videos. In contrast, the accuracy decreased for videos with fast movements, falling below 70%. Overall, the model evaluation achieved promising results, yielding accurate predictions from 0.5 to 1 s, facilitating smoother playback by pre-fetching predicted frames in advance [24].

The method presented in [24] has been integrated into a 360 video streaming framework in [3] to enhance the video delivery quality. The proposed method has been evaluated using many metrics: (i) the buffer stall count, which represents the number of times that

video playback is interrupted due to buffer depletion; (ii) the buffer stall duration, defined as the total number of seconds for which playback is halted while waiting for the buffer to refill; and (iii) the blank ratio or missing ratio. Moreover, the Structural Similarity Index (SSIM) has been used to measure the differences between tiles of the predicted viewport and the ground truth. Additionally, they measured the saved bandwidth. For comparison, the model was evaluated against three baseline methods. The first [71] used linear regression to predict the viewport by providing as input to the model the saliency map and head orientations. The second baseline performed prediction using only the saliency map, excluding the head orientation. The third baseline skipped prediction entirely and simply streamed all tiles as the output. The experimental evaluation demonstrates that the proposed model reduced bandwidth consumption by 31% with respect to the baseline approaches, while simultaneously reducing the stall duration by 64% and the stall count by 46%. These improvements significantly enhance the QoE in 360-degree video streaming systems.

Li et al. contributed to viewport prediction through LSTM by proposing a model consisting of two components [25]. The first component operates offline and is responsible for generating head maps, while the second component works online to predict future viewports for the target user using an LSTM network. In the first component, head maps are generated by combining the results from saliency estimation and motion detection. Saliency estimation employs the Fused Saliency Map (FSM) method proposed in [72], which is an adaptation of existing saliency models for 360-degree images. Motion detection utilizes Lucas–Kanade optical flow [62]. In the second component, the LSTM network predicts tile probabilities using the head maps generated by the first layer, along with the previous head orientations and navigation speed of the target user. In addition, a correction module is implemented on the LSTM network output, applying a threshold to determine which tiles will be in the future viewport. The LSTM network produces two outputs: a short-term prediction for the next second and a long-term prediction for the next 2 s. The method has been evaluated on the dataset presented in [46], using the accuracy and F1-score as performance metrics. The authors also compared the proposed model with the regression-based approach presented in [10]. The comparison showed improved accuracy, with an average of 85.11% compared to 72.54% in [10] on the validation set. Additionally, the proposed method demonstrated more stability compared to [10].

In a different direction, Xu et al. introduced a model for gaze prediction in 360-degree videos, along with a large-scale eye tracking dataset specifically designed for dynamic 360-degree videos [26]. This model utilizes previous gaze data with saliency and motion features to forecast the displacement of the gaze points. The model includes three modules: (i) the trajectory encoder module, (ii) the saliency encoder module, and (iii) the displacement prediction module. The former employs an LSTM network to encode past eye gaze paths, which are crucial in predicting gaze point displacement. The saliency encoder module processes saliency maps and motion maps across multiple scales, utilizing Inception-ResNet-V2 [73] to extract features for gaze prediction. Saliency is calculated at three scales: local, FoV, and global. Local saliency is computed through the inner product between the 360-degree frame and a Gaussian window centered at the current gaze point. FoV saliency is generated by providing as input the current viewport to SalNet [69], a saliency detection method designed for 2D images. Global saliency is obtained by processing the entire 360-degree frame through SalNet. Motion maps are computed following a similar procedure across three scales, local, FoV, and global, using FlowNet2 [74]. The displacement prediction module utilizes the outputs from the trajectory encoder and saliency encoder modules, employing two fully connected layers to predict the displacement between the current and future gaze points. For performance evaluation, Xu et al. set the

prediction horizon to 1 s, training and testing the model on a custom dataset. The dataset consisted of 208 360-degree videos, each viewed by 45 users wearing an HTC VIVE headset with the "7invensun a-Glass" eye tracker. The dataset encompassed diverse dynamic content, including indoor and outdoor scenes such as sports games, music shows, short movies, and documentaries. Videos were captured using both fixed and moving cameras. The proposed method has been compared with four baseline approaches: one that predicts the viewport based on the most salient point, another that uses optical flow to estimate the gaze direction, a third that relies only on the saliency encoder, and a fourth that uses only the trajectory encoder. Performance was evaluated using the mean intersection angle error (MIAE) across all videos and users. Xu et al. demonstrate that the proposed method outperforms the baselines, showing the effectiveness of their proposed model in predicting gaze point displacement.

LSTM networks have been employed also in [30] to predict the future PoIs in 360-degree videos. The authors observed that users tend to focus on a Point of Interest (PoI) after an initial exploration phase. Utilizing six existing datasets, they selected 31 videos with at least one PoI, manually identifying PoIs and comparing them with user-viewed viewports. This was conducted to determine if users followed the PoI or were in exploration mode, i.e., scanning the 360-degree scene without focusing on a specific point. In a preliminary analysis with 30 users across five videos, they found that users spent an average of 81.44% of the playback time viewing a PoI. For viewport prediction, 25 video chunks of 1 s were provided as input to an LSTM network to predict the next three chunks. During the initial 25 s, the naive approach is used as a viewport prediction method, as the model requires 25 s of data to work. Once the model begins producing its own predictions, each output is evaluated for success or failure. If the failure rate exceeds a predefined threshold, the system reverts to using the naive approach instead of the proposed method. Their proposed viewport prediction achieved approximately 60% accuracy, compared to about 58% accuracy for the naive approach.

Moreover, in [28], a framework for viewport prediction tailored to live streaming scenarios has been proposed. The framework aims at achieving three key objectives: (i) online training and prediction to adapt to dynamically generated live VR content; (ii) lifelong updating of the prediction model throughout the video; and (iii) real-time processing achieved through subsampling to reduce delays. The framework combines Convolutional Neural Network (CNN) and LSTM networks for prediction, leveraging both video content analysis and head orientation trajectory analysis. Initially, only the CNN network operates, as there are no trajectories to train the LSTM. When partial trajectories become available, predictions are based on the combined output of the CNN and LSTM networks. The CNN employs a modified VGG network [63] to classify the input into two categories: interesting and not interesting. To reflect every user's preferences, the CNN model starts to train after the user starts watching the video. The training process is achieved by taking the set of frames that the user has watched and dividing it into patches, which are taken as input by the CNN. To account for fast head movements, the LSTM network utilizes previous head trajectories to predict the tiles belonging to the next viewport. The evaluation of the framework has been performed using the dataset presented in [42], with a prediction horizon of 2 s. The model has been evaluated using three metrics: accuracy, bandwidth usage, and processing time. The processing time for each segment should be less than the segment duration, which was set to 2 s in their work. The model was compared with three baseline approaches. The first baseline used a CNN for viewport prediction, the second utilized an LSTM model for prediction, and the third referred to a model proposed in [75]. The simulation tests showed accuracy of 90%, along with approximately 40% bandwidth savings and fast processing times, meeting the real-time and bandwidth requirements for VR streaming, with accuracy ranging from 97% to 99%.

A comprehensive analysis of LSTM-based viewport prediction has been presented in [29]. For a performance comparison, the naive approach has been first compared to a deep position-only baseline, which exploits only head position information for prediction, utilizing a sequence-to-sequence framework with LSTM units for encoding and decoding. The authors set the prediction horizon to 5 s and initiated predictions after 6 s of video playback to bypass the exploration phase. These approaches were compared to the methods presented in [14,23,24,26,41]. The authors demonstrated that the two baselines outperformed existing approaches for specific types of video content and showed that the methods incorporating saliency information failed to surpass the deep position-only baseline.

To investigate the contribution of saliency information, a saliency-only baseline has been introduced. The ground-truth saliency was derived from tracked eye positions or was set as the center of the explored viewport from the employed datasets. The authors observed the significance of past head positions for short-term prediction and proved their relevance for scenarios with frequent low-velocity and inertia effects present in the videos. In contrast, the use of the ground-truth saliency indicated a notable improvement in viewport prediction for non-exploratory videos, i.e., the ones containing a key PoI that attracts users' attention, particularly for prediction horizons exceeding 3 s. However, for shorter horizons, typically less than 2.5 s, the contribution of saliency remained uncertain. The performed analysis suggests that, when using ground-truth saliency, the optimal approach involves employing a dedicated recurrent unit to process position information before merging it with saliency-based features. Then, the observations derived from ground-truth saliency information were extended to a case in which the estimated saliency was employed. To this aim, the PanoSalNet model [24] was employed. The performed analysis shows that the saliency contribution in viewport prediction decreases when the saliency estimation is not accurate.

As a final contribution, the authors proposed the TRACK model, which combines saliency and head position information in three steps:

1. A double-stacked LSTM with 256 units each, processing head movement information;
2. A double-stacked LSTM with 256 units each, processing content-based saliency after flattening it;
3. A double-stacked LSTM with 256 units each, processing the concatenated output from the first and second steps.

For performance evaluation, the datasets presented in [23,24,26,41,46] have been employed. The proposed model was compared against several baseline methods (i.e., [12,14,23–26,41]). To ensure a fair and comprehensive evaluation, the authors adopted the evaluation metrics used for each respective baseline. Specifically, the accuracy and F1-score were used for comparison with the method in [23]. The mean overlap was used for comparison with the method in [41]. The intersection angle error was employed for comparison with [26], while the IoU was used to evaluate the performance relative to the approach in [24]. As a result, the proposed method achieved a mean overlap of 0.968 on average, significantly outperforming the baseline method in [41], which achieved 0.753 for a prediction horizon of 30 ms. Additionally, for a 1 s prediction horizon, the proposed method attained accuracy of 95.48%, compared to 86.35% reported by [23]. After comparing TRACK with the other state-of-the-art baselines, TRACK demonstrated superior performance with respect to [12,14,23–26,41] for prediction horizons exceeding 3 s. For shorter prediction horizons, TRACK's performance was comparable to that of the deep position-only baseline.

2.3.2. Convolutional LSTM

A possible alternative to standard LSTM is represented by Conv-LSTM [76], which allows the integration of spatial information. In this direction, Dong et al. introduced the gaze-based field of view prediction (GEP) framework [32]. It forecasts long-term

viewports by leveraging the target user's previous eye gaze trajectories together with the future trajectories of other users watching the same video. They represented historical movement trajectories through eye gaze-based heatmaps computed over 10 s of data and selected similar users through dot product operations. Dong et al. noted that the outcome of this process resembles that of saliency detection but is constructed based on real human attention. The proposed method is based on the combination of a seven-layer U-Net [77] integrated with the Squeeze-and-Excitation Network (SE-net) [78] and Conv-LSTM. Initially, the Convolutional Long Short-Term Memory (Conv-LSTM) model performs temporal predictions using the target user's eye gaze heatmap; then, the SE-U-Net network makes the final prediction based on the temporal prediction results and the eye gaze heatmaps of other users. To generate heatmaps, the FoV's center was set to the tile that the eye was fixated on, while other tiles received values based on their distance from the FoV's center. The prediction horizon was set to 10 s, and the evaluation was conducted on the dataset presented in [26], which consists of 208 videos watched by 45 users. To evaluate the proposed model, the authors used a set of metrics to assess the effectiveness of their method. These included the hit rate, which is defined as the average percentage of the predicted FoV area (computed frame by frame) relative to the actual FoV area within a given second; the viewport deviation; and the mean squared error (MSE), which measures the average squared distance between the predicted and actual FoV center positions in a 2D projection. Additional metrics are the viewport PSNR, bandwidth occupation, viewport quality variance, and re-buffering time. The proposed model was evaluated against several baselines, including the method introduced in [12], as well as various ablated versions of the proposed approach, such as configurations using only eye gaze heatmaps for prediction. The authors showed that the proposed method achieved high effectiveness in predicting long-term (10 s) viewports, outperforming other state-of-the-art methods. As an example, the PSNR achieved by the proposed method is 25.535 dB, while the method presented in [12] achieves 22.980 dB. The average re-buffering time (in seconds) of the proposed method is 0.702 with respect to the 0.822 reported in [12].

In addition, in [31], the CoLIVE framework was introduced. It is designed for live streaming scenarios where viewport prediction is performed online using edge computing. A key innovation with respect to previous approaches is that, as many users exhibit similar behaviors when watching 360-degree videos, CoLIVE implements collaborative online prediction by sharing local losses with a central model, aiding new viewers in viewport prediction. In addition, CoLIVE utilizes Conv-LSTM [76] on the edge side rather than the standard LSTM, due to its capability to capture both the spatial structure of frames and temporal features. CoLIVE operates through several steps. First, a saliency detection model generates a saliency map on the server side and transmits it to the edge side. Second, viewport prediction for each chunk is conducted using Conv-LSTM on the edge side, taking saliency maps and previous viewport trajectories as input. The output provides future viewport locations, and the model parameters are updated after the user watches the video chunk. Third, these parameters are shared with a central model to assist new users in predicting their viewports. Saliency estimation is performed using a CNN with eight convolutional layers and two max-pooling layers. Viewport prediction employs Conv-LSTM followed by two convolutional layers and two de-convolutional layers to predict future viewport locations, utilizing saliency maps and previous head trajectories as input. For experimental validation, a prototype system was created and the model was tested with two public datasets [42,79]. The performance of the proposed model was evaluated using several metrics, including accuracy, recall, precision, bandwidth savings, and processing time. The evaluation was conducted in comparison with the two methods presented in [24,28]. Processing time was evaluated for both saliency estimation and

viewport prediction. For viewport prediction, the model successfully predicted the next viewport within the duration of the current video chunk, which was set to 2 s, allowing for smooth and uninterrupted streaming. Wang et al. reported that their framework outperforms the state-of-the-art methods in terms of bandwidth savings and accuracy [31].

Wang et al. later enhanced their model for collaborative online learning by introducing the Dynamic Federated Clustering Algorithm (DFCA), designed to improve the viewport prediction accuracy [33]. This algorithm dynamically groups users based on the gradient updates of their model and the viewing behaviors within a sliding window. The selection of a clustering approach was motivated by the fact that the analysis of data traces revealed consistent viewing patterns among groups of users. By leveraging these similarities, the model can make more accurate predictions for users within the same cluster and provide initial predictions for new users. The DFCA primarily comprises two phases (i.e., dynamic partitioning and periodic merging) to dynamically separate or merge viewer groups based on user behavior by periodically computing the Euclidean distance between their gradient updates. The model was evaluated using the dataset presented in [42]. To evaluate the proposed method, several metrics were employed, including accuracy, precision, recall, the F1-score, and bandwidth savings. The proposed method was compared to LiveDeep [28], PanoSalNet [24], and the earlier CoLIVE model (referred to as CoLIVE w/o) [31]. In terms of average bandwidth savings, CoLIVE w/o achieved the highest value, corresponding to 34.12%, closely followed by the enhanced CoLIVE, which reached 34.04%. In contrast, PanoSalNet and LiveDeep achieved only 28.14% and 21.48%, respectively. The enhanced CoLIVE also demonstrated the highest average accuracy (96.47%), outperforming CoLIVE w/o (95.43%), PanoSalNet (65.74%), and LiveDeep (68.99%).

Conv-LSTM has been also applied to viewport prediction for mobile devices in [34]. Specifically, Conv-LSTM takes two inputs: (i) a saliency map estimated by a graph convolutional network (GCN)-based predictor and (ii) a tile-based map of the gaze trajectory. The latter is built by predicting future trajectories based on historical data and then mapping them to the existing tile. Each tile is allocated a different bitrate depending on its probability of being watched. The authors use the proposed viewport prediction in a larger framework to improve 360-degree video streaming. Performance assessment has been performed using the dataset presented in [11], and the proposed method was compared to three other streaming frameworks, including Flare [11,80,81]. To evaluate the proposed method, three metrics were used: the PSNR to assess video quality against the reference, the inter-chunk PSNR to measure quality consistency between adjacent video segments, and the re-buffering time to evaluate playback smoothness. The performed experimental tests show an increase of up to 17.3% in the perceived subjective quality and a 20.6% bandwidth reduction with respect to the best-performing methods in the state of the art.

In the aforementioned studies, saliency detection was typically performed using 2D convolutional layers on equirectangular frames, thus creating distortion issues due to the projection process. To overcome this issue, in [82], saliency estimation has been explored from a spherical perspective by including the spherical convolutions initially proposed in [83]. Specifically, to achieve convolution on spherical images via their ERPs, local convolution and max pooling are conducted on the tangent plane of the spherical patch. The authors in [82] introduced the SphereU-Net framework, which comprises a U-Net architecture [77] integrated with spherical convolutional layers and spherical max pooling. It takes the equirectangular video frame at time t and the saliency map at time $t-1$ as input, producing the saliency map at time t as output. The experimental assessment was conducted using the saliency dataset presented in [84], and Peng et al. demonstrated that their proposed saliency detection framework outperformed other existing saliency models, such as [24,84], in terms of the Pearson linear correlation coefficient (CC), Kullback–Leibler

divergence (KL), normalized scanpath saliency (NSS), and area under the curve—Judd (AUC-Judd). The key finding presented in [82] is that, by increasing the saliency estimation accuracy, this method might be able to boost the viewport prediction performance.

2.3.3. Gated Recurrent Unit

Another approach to viewport prediction uses gated recurrent units (GRUs), which are a type of recurrent neural network introduced in [85]. Gated Recurrent Units (GRUs) are similar to LSTM in their ability to handle sequential data by learning which information to keep or forget over time. However, GRUs have a simpler structure with fewer parameters, making them easier to train and more efficient while still capturing important temporal patterns.

Based on this, Li et al. tested the usage of GRUs in combination with spherical convolutions in [35]. They focused on a multicast scenario where previous viewports from multiple users were utilized. Viewport prediction involved the fusion of a saliency feature extraction module and an FoV prediction module. Initially, the video frame at time t was processed by the spatial feature extraction module, which comprised multiple spherical convolutional and de-convolutional layers to identify PoIs in the frame. Concurrently, the temporal feature extraction module analyzed consecutive frames from time t to time n (where n is the prediction period) to extract temporal features. The results of these modules were then fed into the Convolutional Block Attention Module (CBAM), akin to the work done by Woo et al. [86], to enhance the saliency detection accuracy. In parallel, a set of FoV maps of other users is fed into a Spherical Convolutional Gated Recursive Unit (SP-ConvGRU) leveraging spherical convolutions. The outputs of the two parallel branches are combined to provide the predicted FoV. The prediction horizon was set to 2 s. For experimental evaluation, they employed the datasets presented in [26,84]. To evaluate the proposed model, three metrics were used: accuracy, precision, and recall. To evaluate the saliency estimation model and viewport prediction, the model was compared against five baseline methods presented in [11,24,87–89]. Through experimentation, Li et al. demonstrated the superior performance of their module compared to other baseline methods [35].

GRUs have been employed also by Setayesh et al. in [36], who expanded the viewport prediction model they presented in [90]. They propose a hybrid approach that encompasses three elements: a saliency estimation method, a technique for head movement prediction, and an integration approach. Concerning saliency estimation, the authors resort to PAVER [91], which processes ERP video patches through an encoder (i.e., a deformable convolution layer) and a decoder that leverages local, spatial, and temporal information to return the final estimated saliency map. The head movement prediction module is composed of an encoder and a decoder, containing four GRU layers each. The maps obtained in each of these steps are integrated through a regional fusion technique. The experimental evaluation has been performed based on the dataset presented in [84]. To evaluate the performance of the model, the tile overlap has been used. It is defined as the intersection between the predicted and the ground-truth viewport divided by the ground-truth viewport. To assess the QoE, a score based on four key factors was defined. The factors are defined as follows: the average quality of tiles within the viewport, the spatial quality smoothness among these tiles, the temporal quality consistency, and the re-buffering delay. Tile quality was determined based on the bitrate and quantization parameters. The proposed method was compared against the approaches presented in [92–94]. It shows improvements, achieving 2.41% higher overlap compared to [93] and 15.93% higher than [92]. For the QoE, the proposed method outperformed [92–94] in terms of the defined score by 23.77%, 62.7%, and 116.57%, respectively.

2.3.4. Transformers

As an alternative approach for viewport prediction, Transformers have been employed in [37] due to their advantages in handling time-series problems and capturing long-term dependencies compared to RNN networks [56,95]. Specifically, the authors introduced the VRFormer network to address challenges in FoV prediction and adaptive video streaming. The primary challenge is that the streaming quality of 360-degree videos relies on accurate FoV prediction, as lower prediction accuracy leads to wasted bandwidth and poor user experience. The second challenge arises when dealing with tile-based 360-degree video streaming, where the bitrate needs to be dynamically adjusted for different tiles based on the network environment and current video player. To deal with the first challenge, the proposed Transformer takes as input video content, previous head movements, and eye traces. For the second challenge, a DRL network is employed to model the reward for improved user QoE, dynamically controlling video content reconstruction and adaptively allocating rates for future tiles. Focusing on FoV prediction, the VRFormer network consists of three main processes: spatial feature extraction, temporal feature extraction, and content-aware FoV prediction. Spatial features are extracted by dividing 360-degree video frames into 2D FoV images and using a pre-trained ResNet-50-based CNN [96] to extract spatial features and generate attention feature maps. Temporal dependencies are captured using a Transformer encoder network, which takes into account the history of head motion and eye tracking from the initial frame to the current one. The results from both processes are then fed into a Transformer decoder network, which in turn provides its output to three fully connected layers to predict future viewports. The experimental evaluation has been performed on the dataset presented in [26], with the prediction horizon set to 10 s. To evaluate the viewport prediction performance, two metrics were used: the viewport overlap ratio and the average MSE between the predicted and actual viewport regions. The viewport overlap ratio has been computed here as the area of intersection between the predicted and ground-truth viewports divided by their union. The methods presented in [12,23,97,98] were used for comparison. To evaluate the proposed approach in terms of bitrate adaptation and QoE, three additional methods presented in [53,81,99] were considered. The QoE was assessed using three criteria: (i) the average video quality, measured by the average bitrate of tiles within the viewport; (ii) the video smoothness, reflecting fluctuations in tile quality inside the viewport; and (iii) the re-buffering time, which captures playback interruptions. The comparison demonstrates that the proposed approach offers better user QoE compared to other state-of-the-art methods. As the results show, the predicted viewport overlap declines progressively with the extension of the prediction horizon: from 95% at 1 s, to 75% at 5 s, and down to 73% at 10 s. This degradation reflects the increasing difficulty in accurately predicting the viewport over time.

Transformers have been employed for viewport prediction also in [38]. The authors combine historical viewpoint trajectories, which should account for short-term viewing behaviors, and saliency information, which should provide a crucial contribution for long-term predictions. First of all, they process the video frames with the PAVER saliency estimation technique [91] and then convert the obtained saliency to a compact 2D representation that encodes the 3D spatial coordinates and the saliency value. In parallel, an LSTM module uses the past viewpoint positions, i.e., the head scanpath, to perform short-term viewpoint prediction. Then, the LSTM output and the saliency map are provided as input to a spatial attention module that aligns the two information types. The aligned saliency and short-term predictions are provided as input to a temporal attention module, which provides a long-term prediction. Finally, the long- and short-term predictions are fused through a gating fusion module. Performance has been assessed using the datasets presented in [42,45,51]. For performance evaluation, the IoU and orthodromic distance

have been used. The proposed approach has been compared with LR, with a custom LSTM-based encoder–decoder module, as well as state-of-the-art approaches such as the ones presented in [20,29,100]. The comparison highlighted that the regression technique, the LSTM-based approach, and the method in [20] had better performance for short-term prediction (less than 1 s). With respect to them, the approach presented in [29] had better long-term performance (2–5 s), while achieving less accurate short-term predictions. In contrast, the method proposed in [38] achieved the best long-term predictions without resulting in poor short-term accuracy. Specifically, the proposed method achieved the lowest orthodromic distance and the highest IoU on all datasets.

The use of Transformers for viewport prediction has been extended to bidirectional Transformers in [39]. Specifically, the proposed framework has three main components: (i) a multi-level residual module aimed at extracting multi-level features through spherical convolutions; (ii) a bidirectional encoder–decoder module based on Transformers, which takes as input the spatial features and extracts inter-frame spatiotemporal information; (iii) a spherical matching module that predicts the viewport proposal along with the corresponding saliency score in order to provide the output viewport trajectory. Performance assessment has been performed using the datasets presented in [26,41,47] based on three metrics: the IoU, mean overlap, and spherical great circle distance. The method was compared against the approaches presented in [24,41,101–104], showing improved performance. For instance, the model demonstrated its strongest performance on the dataset presented in [41], achieving a great circle distance of 0.341, an IoU of 0.667, and a mean overlap score of 0.751. In comparison, on the dataset presented in [47], the model reached a great circle distance of 0.463, an IoU of 0.588, and a mean overlap score of 0.677. Lastly, on the dataset from [26], it achieved a great circle distance of 0.510, an IoU of 0.560, and a mean overlap score of 0.652.

2.3.5. Graph Neural Network (GNN)

Another valuable approach for viewport prediction is presented by graph neural networks (GNNs) [105]. In [40], a Graph Neural Network (GNN) is employed to forecast future viewports by fusing multiple features. Specifically, a sparse direct graph is built using three extracted features: past head orientation traces of the target user, the video saliency and motion maps, and the head orientation traces of other users viewing the same video. Past head orientations were processed through a two-layer LSTM to provide the head orientation prediction. Salient regions were identified using SalNet, a method tailored to saliency detection in 2D images [69]. For motion maps, FlowNet [74] was employed. For cross-user heatmap generation, Xu et al. relied on the K-means clustering algorithm to group users exhibiting similar behaviors while watching the same video. To align the dimensions of this output with other components, a tile-level map is generated from the predicted orientations. As for the other users' information, a cross-user heatmap was generated by dividing the video into tiles and computing the normalized probability for each tile. These three categories of features have been employed to define the directed graph. The number of tiles corresponds to the number of nodes in the graph, whose weight is defined as a function of the extracted features, while the edges represent the associations between the tiles. The graph is then provided as input to a two-layer GNN whose output indicates the viewing probability for each tile. The prediction horizon was set to 3 s and the model performance was evaluated using the dataset presented in [46]. To evaluate the proposed method, accuracy was used as the primary metric for comparison with other state-of-the-art approaches. Additionally, the QoE was assessed based on three key factors: the total quality of tiles within the viewport, the spatial quality variation among these tiles, and the temporal quality variation between two consecutive video segments. In this evaluation, tile quality was determined based on the total bitrate. The proposed method

has been tested against the approaches presented in [28,106]. The authors showed that the proposed approach outperforms others across various time windows. Specifically, the proposed model achieves average accuracy of 84.6% for a 1 s prediction window, outperforming the best baseline method presented in [28] by 11.1%. Even under a longer 3 s window, the method maintains accuracy of 68.7%. The authors highlight the model's superiority over existing methods, demonstrating its versatility and adaptability across various video types.

2.3.6. Reinforcement Learning (RL)

Furthermore, reinforcement learning (RL) approaches have been proposed for viewport prediction. For instance, Xu et al. employed DRL to forecast head movement positions, introducing a method called deep reinforcement learning-based head movement prediction (DHP) [41]. Deep Reinforcement Learning-based Head Movement Prediction (DHP) operates in both offline and online modes. The output of offline-DHP is a set of potential head positions, generated by running multiple DRL workflows to determine potential head movement positions in each 360-degree frame. This offline-DHP scheme is designed to model the attention of multiple users. In contrast, the online-DHP model predicts the next head position for an individual based on current and previous frame head movements in real time. The observed content of the 360-degree videos serves as the environment, while the magnitudes and directions of head movement scanpaths serve as actions for multiple DRL agents. The reward is based on the differences in actions between subjects and agents, reflecting how closely the agents mimic the subjects' scanpath predictions. The agent selects the appropriate scanpath action for the next frame based on the ground truths of previous head movement scanpaths and content observation. Online-DHP emphasizes attention to content by leveraging the learned offline-DHP model. Both the offline-DHP and online-DHP networks are structured with one LSTM layer and four convolutional layers, aimed at extracting temporal and spatial content features. The online-DHP network predicts head positions for the frame at $t + 1$ by utilizing the head movements of target users from time 0 until t as input. In their experiments, Xu et al. developed the PVS-HM dataset, comprising 76 360-degree videos watched by 58 users with the HTC Vive HMD, including head movement data and eye fixations captured by the embedded eye-tracking module. The proposed method was evaluated using the mean overlap as the primary metric. It was tested against several other techniques, including the naive approach, a method that predicts the viewport randomly, and the method presented in [107]. The authors showed that their model achieved superior performance over other approaches, such as [107], in both online and offline head movement prediction tasks. Specifically, online-DHP predicts the user's future head position for the next frame with an average mean overlap of 0.75, outperforming the method presented in [107], which achieves 0.40, and the naive approach, which obtains a much lower mean overlap score, i.e., 0.20.

3. Datasets

All 360-degree videos can be classified into different categories based on the visual content [29,108]. In turn, the video content influences how the user explores the 360-degree scene, thus impacting also viewport prediction. The main video categories are reported in the following.

- Exploration videos: In this category, users tend to explore the entire scene without focusing on a specific object or region. These scenes usually contain no motion and no dominant visual target. An example is represented by natural landscapes or panoramic views. As a result, while watching the video, head movements are widely distributed, thus making viewport prediction particularly challenging.

- Static focus: These videos contain salient objects that remain fixed near the center of the scene, such as a performer in a concert or a static speaker. Because the main object naturally attracts attention, these videos are considered easier for viewport prediction tasks.
- Moving focus: In these videos, salient objects move around the scene, requiring users to follow their motion. An example of this category is represented by sports videos, where multiple players move inside the scene, or dynamic scenes with multiple actors. Predicting head orientation is more complex because of the variability in motion and attention shifts.
- Ride-based videos: This category includes camera motion during video capture. One of the most common scenarios is represented by car driving. These videos often introduce a forward-moving visual flow, which adds another layer of complexity to predicting user gaze and head movement.

Among these video categories, exploration videos are considered the most difficult for viewport prediction due to the lack of obvious visual cues and motion, while static focus videos are the most predictable due to their centered, low-motion content. Additionally, the viewport prediction accuracy can vary depending on the type of content or the dataset used. Some videos begin with an impactful scene, such as an explosion, after which the real content appears. Because of this, some viewport prediction methods have resorted to cutting the first six or ten seconds to achieve accurate viewport predictions; an example is provided in [29]. Another possibility consists in filtering the dataset to include only the videos that have at least one PoI, as described in [30].

As demonstrated in the previous sections, several publicly available datasets have been used for the training and evaluation of viewport prediction and saliency estimation models. We provide a detailed discussion of the three largest datasets, referred to as VR-EyeTracking, Sport360, and PSV-HM, in the following.

- VR-EyeTracking—introduced in [26]: This is the largest known 360-degree video dataset, which contains 208 videos with a 4 K resolution; each video has been watched by 45 users. The videos in the dataset have durations ranging from 20 to 60 s at 25 fps. The video content is varied, including indoor scenes, outdoor environments, sports, music performances, and documentaries. Users watched the videos using an HTC Vive headset with a 7invensun a-Glass eye tracker. This dataset is provided in the form of MP4 videos, along with the corresponding head and eye fixations.
- Sport360—introduced in [84]: This dataset includes 104 videos (extracted from the Sport-360 dataset [107]) viewed by 27 participants. As for the first dataset, the HTC Vive and the a-Glass eye tracker were used. Differently from VR-Eyetracking, this dataset provides just the video frames and the corresponding saliency maps along with the ground-truth eye fixations, making it useful for saliency estimation tasks.
- PSV-HM—introduced in [97]: This dataset contains 76 videos viewed by 58 users. The videos, taken from YouTube and VRCun, have a duration of between 10 and 80 s and resolutions varying from 3K to 8K. They were watched using the HTC Vive, with the head orientation recorded for each viewer. The dataset includes diverse content types, such as animation, gaming, sports, nature scenes, and driving. It is available as MP4 videos along with the corresponding head orientation data.
- Other datasets: Additional datasets include collections with a diverse number of videos and subjects, offering varied levels of annotation and content diversity.

We provide in Table 2 a summary of the datasets used for the viewport prediction task. For each dataset, we report (i) the number of videos that it contains, (ii) the number of users who viewed them, (iii) the video duration in seconds, (iv) the video resolution, (v) the source from which the videos were obtained, (vi) the modality used for tracking head

and/or eye movements during the experiments, (vii) the studies or methods in which the dataset was employed, and (viii) an indication of whether the dataset is publicly available online. This overview helps to illustrate the scale and diversity of data used across the viewport prediction methods included in this review.

Table 2. Dataset summary. The dash (-) indicates that the information was not available in the reference paper.

Reference	Videos	Users	Duration	Resolution	Video Source	Data Capture System	Employed In	Available
Qian et al. 2016 [10]	4	5	100–206 s	-	YouTube and Facebook	Head orientation from smartphone inside headset	[10]	no
Bao et al. 2016 [9]	16	153	30 s	1080 p–4 K	YouTube	Head orientation from HMD	[9,14,15]	no
Wu et al. 2017 [42]	18	48	164–655 s	-	-	Head position and orientation from HMD	[12,20,22,28,31,33,38]	yes
Fan et al. 2017 [23]	10	25	-	4 K	YouTube	Head position and orientation from sensor logger and screen capturer	[17,23,29]	no
Lo et al. 2017 [46]	10	50	60s	4 K	YouTube	Head position and orientation from sensor logger and screen capturer	[8,17,25,27,29,40]	yes
Qian et al. 2018 [11]	10	130	117–293 s	4 K	YouTube	Head orientation from smartphone inside headset	[11,34]	yes
David et al. 2018 [51]	19	57	20 s	4 K	YouTube	Eye tracker and head scanpath (derived from eye)	[19–21,38]	yes
Nguyen et al. 2018 [24]	11	48	20–45 s	-	Videos from existing datasets	Eye gaze from head orientation logs in original datasets	[3,17,24,29]	yes
Xu et al. 2018 [26]	208	45	20–60 s	4 K	YouTube	Eye tracker	[17,26,29,32,35,37,39]	yes
Zhang et al. 2018 [84]	104	27	20–60 s	-	Videos from existing datasets	Eye tracker	[35,36,82]	yes
Xu et al. 2019 [41]	76	58	10–80 s	3–8 K	YouTube and VRcun	Head orientation form HMD and eye tracker	[17,20,29,39,41]	yes
Li et al. 2018 [47]	60	221	10–23 s	4–8 K	Custom and YouTube	Head orientation form HMD and eye tracker	[18,39]	yes
Nasrabadi et al. 2019 [45]	28	30	60 s	2–4 K	Custom (Samsung Gear 360) and YouTube	Head orientation from HMD	[16,38]	yes
Nguyen et al. 2019 [79]	24	48	60–655 s	-	Videos from existing datasets	Eye gaze from head orientation logs in original datasets	[31]	yes
Manfredi et al. 2023 [30]	31	30	-	-	Manual PoI extraction from existing datasets	Actual PoIs from extracted fixations	[30]	no

4. Evaluation Metrics and Comparison Methods

In this section, we review the metrics used to evaluate the performance of viewport prediction methods and provide a summary of how the papers discussed in the previous sections compare their results with the state of the art.

4.1. Evaluation Metrics

Several metrics have been employed to evaluate viewport prediction methods, with variations across the approaches. Commonly used metrics include the accuracy, F1-score, mean FoV overlap, and IoU, among others.

Some studies have also integrated viewport prediction into streaming scenarios, where the QoE is assessed using a range of indicators. These include the average bitrate within the user's viewport for each frame (referred to as Q1), the variation in bitrate within the viewport per frame (Q2), bitrate fluctuations across consecutive chunks or segments (Q3), and bitrate variation across frames within a one-second interval (Q4). Additional metrics such as the re-buffering time and buffer stall count are also used. In addition, some methods

consider network-related factors such as bandwidth consumption, the peak bandwidth, and bandwidth savings.

In Table 3, we list all state-of-the-art methods discussed in this work, along with the evaluation metrics that they employed and the comparison methods used to benchmark the proposed approaches. The employed metrics are detailed below.

- **Accuracy:** the ratio of correctly classified tiles to the total number of tiles.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} = \frac{\text{TP} + \text{TN}}{\text{TT} + \text{TN}}, \tag{1}$$

  where true positive (TP) indicates the tiles predicted to be in the viewport and actually in the user's viewport, and true negative (TN) denotes tiles predicted to be outside the viewport and actually not in the user's viewport. False negative (FN) indicates the tiles in the user's viewport that were not predicted. Finally, false positive (FP) indicates the tiles predicted to be in the viewport but not in the actual user's viewport. Total true (TT) represents the sum of TP, FN, and FP.
- **Intersection over Union (IoU):** the intersection between predicted tiles and ground-truth tiles divided by the union of predicted and ground-truth tiles.

$$\text{IoU} = \frac{\text{TP}}{\text{TT}} = \frac{|\text{Pred} \cap \text{GT}|}{|\text{Pred} \cup \text{GT}|}. \tag{2}$$

- **Recall:** answers the question of how many tiles predicted by the system the user actually looked at.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{|\text{Pred} \cap \text{GT}|}{|\text{GT}|}. \tag{3}$$

- **Precision:** answers the question of how many of the tiles that were predicted to be in the user's viewport actually were in the viewport.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{|\text{Pred} \cap \text{GT}|}{|\text{Pred}|}. \tag{4}$$

- **F1-Score:** the harmonic mean of precision and recall.

$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

- **Mean FoV Overlap:** the average ratio of the IoU of the predicted and ground-truth viewport areas over a defined prediction window. The overlapping area is computed as

$$\text{FoV overlap} = \frac{\text{Area}(\text{Pred} \cap \text{GT})}{\text{Area}(\text{Pred} \cup \text{GT})} \tag{6}$$

- **Intersection Angle Error:** the distance between the predicted point and the ground-truth point on a sphere.
- **Great Circle Distance/Orthodromic Distance:** the shortest distance between the center of the predicted viewport and the ground-truth viewport on the surface of a sphere.
- **Average Great Circle Distance:** the average of the great circle distance over all future time steps.
- **Viewport PSNR(V-PSNR):** where the PSNR is computed only over the user's viewport compared to the reference ground truth.
- **VWS-PSNR:** where the PSNR is computed over the user's viewport, incorporating a spherical correction and the weighted MSE compared to the reference ground truth.

- **Hit Rate:** the average percentage of the predicted FoV area, computed frame by frame within a given second, relative to the total area of the actual FoV across those frames.
- **Missing Ratio/Blank Ratio:** the ratio of missing tiles to viewed tiles.
- **Unseen Ratio:** the ratio of tiles that arrived at the client but were not watched to the total number of transmitted tiles.
- **Bandwidth Consumption:** the amount of bandwidth used to stream the predicted tiles.
- **Peak Bandwidth:** the highest bandwidth consumption recorded during the streaming process.
- **Total Re-Buffering Time:** the overall time spent re-buffering during playback.
- **Buffer Stall Count:** the number of times that video playback is interrupted due to buffer depletion.
- **Buffer Stall Duration:** the total number of seconds for which playback is halted while waiting for the buffer to refill.
- **Viewport Deviation:** the percentage of blank area inside the viewport.
- **Viewport Quality Variance:** adopts the coefficient of variation (CV) to evaluate the viewport quality variance in one segment (as detailed in Section 2.1.1).

We analyzed the use of these metrics across the studies and identified the three most commonly employed ones (reported in bold in Table 3): accuracy, mean FoV overlap, and re-buffering time. Notably, while the first two directly evaluate prediction performance, the re-buffering time reflects user annoyance, thus capturing the final user experience.

Despite this, in general, the high variability in the metrics employed to evaluate a viewport prediction approach makes a performance comparison between different approaches challenging. This is further complicated by the fact that some ambiguities in metric definitions exist across studies. Specifically, the ambiguities involve the definitions of the accuracy, mean FoV overlap, and IoU, for which we provide a detailed discussion in the following.

4.1.1. Accuracy

Accuracy is interpreted differently across studies. In this work, we use the term accuracy whenever it is employed in the original papers, ensuring a coherent discussion. The specific accuracy definitions are reported in the following.

In some works, it is clearly defined as the number of overlapping tiles between the predicted and the ground truth, divided by the union of predicted and actual ground-truth tiles—essentially the same as the IoU. This definition is used in [24,35,40]. In contrast, other studies define the prediction accuracy as the number of accurate predictions divided by the total number of predictions, as seen in [10,11].

Several papers define accuracy as the ratio of correctly classified tiles to the union of predicted and ground-truth tiles. However, they do not specify whether "correctly classified" includes both TP and TN or only TP. This interpretation appears in [17,23,25,29]. In [28], a stricter definition is provided: a prediction is deemed accurate only if the predicted viewport completely covers the ground-truth viewport. The final accuracy is then the ratio of correctly predicted frames to the total number of frames. Finally, some works (e.g., [30,31,33]) mention accuracy without offering an explicit definition.

Based on the general definition of accuracy, as well as to differentiate it from the IoU, we recommend the definition provided in Section 4.1.

4.1.2. Mean FoV Overlap

This metric typically reflects the average ratio of the intersection over union of the predicted and ground-truth viewport areas over a defined prediction window. This definition is used in [16,17,20,29]. Other studies use the term "tile overlap" to refer to the ratio between the intersecting tiles divided by the total tiles in the ground-truth viewport, as defined by [36]. Finally, in [37], the authors introduce the concept of the "viewport overlap ratio", calculated per second by averaging the overlap across all frames within that interval. In this review, we employ the metric names and definitions used in the original papers.

4.1.3. IoU

The IoU is another commonly used metric, defined as the ratio of true positive tiles to the total number of unique tiles labeled as part of either the predicted or ground-truth viewport. Thus, the IoU equals the intersection divided by the union of predicted and actual viewport tiles. While similar to the mean FoV overlap, the key distinction lies in its representation: the mean FoV overlap typically deals with continuous spatial areas, while the IoU operates on discrete labeled tiles. The IoU has been used in works such as [17,29,38,39].

Based on the set of metrics employed in previous studies, we propose an evaluation methodology based on three assessments: (i) the performance of the prediction algorithm, (ii) the impact of viewport prediction on network usage, and (iii) the impact of viewport prediction on users' QoE. For the first one, a metric that measures how much the predicted viewport overlaps with the ground-truth viewport has to be selected. Based on their wide diffusion and their similar meanings, we recommend the accuracy, the IoU, and the mean FoV overlap. The second assessment should be performed to quantify the savings in term of network resources. A straightforward metric is represented by bandwidth consumption. The last assessment should relate the streaming performance with the users' QoE. Currently, a unified score does not exist, but a metric could be defined by combining the information concerning the average and standard deviation of the bitrate within the viewport per frame, the bitrate fluctuation across frames or chunks, and buffering stall occurrence and duration. The score should be validated through the use of varied subjective experiments to assess users' perceived QoE.

4.2. Comparison Between Methods

In evaluating their models, many state-of-the-art papers use variations of the proposed approach as a baseline, meaning that they modify internal components or input types (e.g., using only saliency maps or only head orientation history) to assess the performance. The naive approach is another common comparison baseline, which assumes that the future head orientation remains the same as the last observed one. Many methods benchmark their models against this approach.

Table 3. Evaluation metrics and comparison methods; metrics in bold are the top three most used ones.

Reference	Evaluation Metric	Comparison Method
Qian et al. 2016 [10]	**Accuracy**	Variations of own method
Bao et al. 2016 [9]	Mean error, RMSE, 99th percentile, and 99.9th percentile	Naive approach and variations of own method
Qian et al. 2018 [11]	**Accuracy**	Naive approach
Ban et al. 2018 [12]	Viewport deviation, V-PSNR, viewport quality variance, and bandwidth consumption	Variations of own method
Petrangeli et al. 2018 [15]	Great circle distance and **mean FoV overlap**	Naive approach, [12,43]
Nasrabadi et al. 2020 [16]	**Mean FoV overlap**	Naive approach and variations of own method
Dziubinski et al. 2024 [8]	Average bitrate of video overlapping with viewport, Q3, and Q4	Variations of own method
Li et al. 2023 [17]	**Accuracy**, F1-score, **mean FoV overlap**, intersection angle error, and IoU	[23–26,29,41,55]
Mahmoud et al. 2024 [18]	VWS-PSNR	[48–50]
Feng et al. 2025 [19]	Q1, Q2, Q3, and **re-buffering time**	[11,52,53]
Chao et al. 2021 [20]	Average great circle distance, average ratio of overlapping tiles, and **mean FoV overlap**	[16,24,29]
Wang et al. 2024 [21]	Great circle distance, average great circle distance	[20,29]
Aladagli et al. 2017 [14]	Cross-correlation	-
Alice et al. 2024 [22]	Q1, Q2, and Q3	[24]
Fan et al. 2017 [23]	**Accuracy**, F1-score, missing ratio, bandwidth consumption, **re-buffering time**, PSNR, SSIM, PEVQ, and running time	Naive approach, [65], and variations of own method
Nguyen et al. 2018 [24]	**Accuracy**	Variations of own method
Li et al. 2018 [25]	**Accuracy** and F1-score	[10]
Xu et al. 2018 [26]	Mean intersection angle error	Variations of own method
Fan et al. 2020 [27]	Missing ratio, unseen ratio, bandwidth consumption, peak bandwidth, V-PSNR, and **re-buffering time**	Naive approach and DR
Feng et al. 2020 [28]	Bandwidth consumption, accuracy, and processing time	[75] and variations of own method
Rondón et al. 2022 [29]	**Accuracy**, F1-score, **mean FoV overlap**, intersection angle error, and IoU	[12,14,23–26,41]
Nguyen et al. 2023 [3]	Buffer stall count, buffer stall duration, blank ratio, SSIM, and bandwidth saving	[71], stream all tiles (no prediction), and variations of own method
Manfredi et al. 2023 [30]	**Accuracy**	Naive approach
Wang et al. 2022 [31]	**Accuracy**, recall, precision, bandwidth saving, and processing time	[24,28]
Dong et al. 2023 [32]	Hit rate, viewport deviation, MSE, viewport PSNR, bandwidth consumption, viewport quality variance, and **re-buffering time**	[12] and variations of own method
Wang et al. 2024 [33]	**Accuracy**, precision, recall, F1-score, and bandwidth saving	[24,28,31]
Zhang et al. 2025 [34]	PSNR, inter-chunk PSNR, and **re-buffering time**	[11,80,81]
Li et al. 2023 [35]	**Accuracy**, precision, and recall	[11,24,87–89]
Setayesh et al. 2024 [36]	Tiles overlap, Q1, Q2, Q3, and **re-buffering time**	[92–94]
Zhang et al. 2022 [37]	Viewport overlap ratio, MSE, Q1, Q2, and **re-buffering time**	[12,23,53,81,97–99]
Gao et al. 2024 [38]	IoU and orthodromic distance	[20,29,100]
Guo et al. 2024 [39]	IoU, **mean FoV overlap**, and great circle distance	[24,41,101–104]
Xu et al. 2023 [40]	**Accuracy**, Q1, Q2, and Q3	[28,106]
Xu et al. 2019 [41]	**Mean FoV overlap**	Naive approach, random viewport prediction, and [107]

5. Discussion and Recommendations

Over the last few years, viewport prediction techniques have evolved from simple ML models to more advanced DL approaches. This shift has expanded the prediction horizons and enlarged the set of input types, including head orientation, saliency maps, and motion maps.

A key issue to analyze is the complexity of the proposed solution. Indeed, the computational requirements of viewport prediction are critical, as the model needs to be lightweight enough for streaming scenarios and ensure timely predictions. Many approaches take this into account by evaluating the proposed methods in terms of the processing time and re-buffering time, as shown in Table 3. However, several approaches fail to mention this aspect, focusing only on the effectiveness of viewport prediction. In the following, we provide a summary of the different approaches balancing these two aspects.

In the early stages, LR was used for viewport prediction due to its simplicity and low computational overhead. It showed strong performance in short-term prediction, especially within 0.1–0.5 s and up to 1 s. However, the accuracy significantly dropped for longer prediction horizons. The reason behind this phenomenon is that the correlation between past and future head movements diminishes over longer time gaps, leading to increased prediction errors. To tackle this problem, researchers resorted to clustering techniques based on head movement data from other users watching the same video. However, this approach relies on the availability of data from other users, which may not always be the case. Additionally, the assumption that users' viewing behaviors are similar may not hold. Another issue is that clustering methods often fail to outperform simpler, naive approaches in short-term horizons, such as 1 s, as noted in [15], or 3 s, as mentioned in [16]. Moreover, as highlighted in [16], the effectiveness of the clustering approach depends significantly on the video's content, as the locations of RoIs within the video are crucial.

To address the shortcomings of LR and clustering methods, researchers have turned to Transformers and RNN architectures such as LSTM. These models are well suited for learning sequential patterns and analyzing user head movements over time. Their ability to capture temporal dependencies offers a more robust framework for both short- and long-term viewport prediction tasks compared to previous approaches. It is useful to note that, although Transformers have enhanced capabilities in handling temporal information with respect to other Recurrent Neural Network (RNN)-based models, they also introduce additional complexity.

A major advancement in this field has been the integration of saliency maps with previous head orientation data. This dual-input strategy leverages head orientation to handle short-term predictions while using saliency maps to improve the performance over longer horizons. Supporting this integration, Manfredi et al. [30], in a study involving five videos, each containing at least one PoI, found that users spent on average 81.44% of the time focusing on PoIs. This clearly illustrates the strong influence of visual saliency on user attention and the value of incorporating saliency maps in viewport prediction.

Different strategies have been explored to fuse previous head orientations with saliency maps. Several studies have employed LSTM-based models, but relatively few studies have explored the use of GRUs, Transformers, GNNs or deep reinforcement learning.

To increase the prediction performance with respect to standard LSTM, researchers have investigated more complex architectures exploiting the same principles, such as BiLSTM [17]. However, this approach introduces two notable limitations. First, BiLSTM doubles the number of LSTM layers, using one layer for the forward pass and another for the backward pass, thus increasing the computational complexity. Second, and more critically, BiLSTM relies on future head movement data during training. This makes it less

suitable for real-time applications where future data are not available at inference time, thus limiting its practicality in live streaming scenarios.

Another aspect to consider is that, when the saliency maps are integrated into viewport prediction, LSTM-based models process the saliency maps by first flattening them into 1D vectors. However, this process results in the loss of important spatial and temporal structures inherent in the saliency data. To address this limitation, Conv-LSTM models [31–34] have been proposed, as they can preserve both spatial and temporal features by directly processing the saliency maps in their original form. While this approach improves the effectiveness of processing saliency maps in viewport prediction, it also introduces additional computational complexity, which can be a drawback in resource-constrained or real-time scenarios.

To address this issue, different fusion strategies have been proposed. Some methods combine the previous head orientation with the saliency map before feeding them into the LSTM model, processing them together within the LSTM [24]. Other approaches process the head orientation through an LSTM-based architecture and then fuse the output with the saliency map, without directly processing the saliency map inside the LSTM [26]. As an alternative approach, some authors feed the head orientations and the saliency maps into separate LSTM units. The outputs from both units are then concatenated and fed into another LSTM. This approach has shown better performance compared to others, as reported in [29]. However, it also introduces increased computational complexity due to the use of additional LSTM units.

Another key perspective to consider is the impact of viewport prediction on the user's QoE. Indeed, neglecting the QoE can lead to poor user satisfaction, as even accurate viewport predictions may fall short if issues like latency or re-buffering disrupt the viewing experience. While some state-of-the-art methods incorporate QoE evaluation—using metrics such as the average bitrate within the user's viewport per frame, bitrate fluctuations across consecutive segments, the buffer stall count, and others—many approaches fail to consider the QoE at all. However, assessing the QoE remains essential in validating the robustness and practical effectiveness of viewport prediction methods in real streaming scenarios.

Additionally, one relevant issue to highlight is the lack of standardization in performance assessment. Indeed, many state-of-the-art methods use different baselines for comparison and employ different metrics for performance evaluation, as shown in Table 3. For instance, some state-of-the-art approaches compare their results against a naive baseline or even self-compare by altering the type of input to their models, without adequately comparing them against other state-of-the-art methods. This makes it more challenging to evaluate their effectiveness relative to others. Furthermore, there is significant variation in the evaluation metrics used across different methods. Even when the same metric is employed, its definition may differ from one approach to another, as highlighted in Section 4.1. This inconsistency complicates comparisons between state-of-the-art methods. Moreover, the datasets used can vary significantly from one study to another, as illustrated in Table 2, which poses another challenge for fair comparison. Another factor making comparisons difficult is the wide variety of output formats, as seen in Table 1. All of these limitations contribute to the difficulty in establishing a unified standard for performance assessment.

Another limitation is the small size of the available 360-degree video datasets. As reported in Table 2, the largest dataset includes 208 videos and was released in 2018. This restricts the generalization capabilities of models, especially those based on DL. The video content in the datasets is also a very important factor to consider for viewport prediction. Depending on the content characteristics, the prediction accuracy can vary significantly. For instance, Nguyen et al. [3] demonstrated that their model achieved approximately 85%

accuracy on static videos (with minimal movement), but the accuracy dropped to 70% when tested on ride videos featuring fast motion. As the video subject influences users' behavior and might have a direct impact on the performance of the models, having a varied dataset is essential to obtain robust models.

Another challenge to tackle is the lack of effective saliency estimation models for 360-degree videos. As highlighted by the current review, many methods still use 2D saliency models, which do not take into account the distortions typical of 360-degree content. Since saliency maps are often used as input in viewport prediction, improving the performance of saliency estimation models for 360-degree videos is an important research direction. The definition of new approaches specifically tailored to 360-degree content would be a valuable direction in terms of enhancing saliency estimation.

Furthermore, another relevant issue to consider is the limited prediction horizon. Many methods predict viewports for intervals from 1 to 3 s (as shown in Table 1). This short duration is often insufficient for buffering or smooth playback. Although some techniques are designed to reach longer prediction horizons, like 5 or 10 s, using clustering techniques or hybrid approaches, the accuracy of these methods degrades rapidly when the prediction horizon increases, even with advanced frameworks like Transformers. For instance, in [37], a Transformer-based model achieved an overlap ratio of 95% for a 1 s prediction horizon, but this dropped to 75% at 5 s and further declined to 73% at 10 s. These results highlight that, despite architectural advances, accurate long-term viewport prediction remains an open problem requiring further research and innovation. Indeed, while long-term prediction is needed, the model's accuracy has to remain stable over time.

Finally, the size of the input window (e.g., the length of head trajectories or the number of video frames) is a key parameter to set. While the window size must be large enough to detect relevant motion patterns, it should not include noise or unrelated information.

Based on the survey carried out in this work, we provide the following recommendations:

- **Collect rich datasets**: There is a need for larger and more varied datasets that include different content types and user data (e.g., eye gaze, head traces, and saliency ground truth). These will aid in model generalization and their application to real-life scenarios.
- **Unify the evaluation procedure**: Shared evaluation datasets and metrics would be a relevant contribution. This will allow fair and objective comparisons between different approaches.
- **Design efficient models**: Future research should focus on lightweight models that are suitable for real-time deployment on low-end devices. In addition, scalability should be granted for multi-user and long-term prediction scenarios. Furthermore, models should be developed with a user-centric perspective by incorporating QoE as a core evaluation criterion—ensuring not only the accuracy of viewport prediction but also smooth playback, reduced latency, and enhanced overall viewing satisfaction.
- **Improve saliency estimation**: It is important to design robust saliency estimations for 360-degree videos. They should account for spherical distortions and large frame sizes.
- **Optimize input windows**: The input data size should be carefully set in order to balance the input quality, the risk of noisy samples, and high computational complexity.
- **Investigate diverse input types**: Current approaches mainly rely on video content and users' behavioral data. Future research could explore combining these with physiological signals.
- **Realize QoE by-design**: While QoE metrics are often used for model evaluation, QoE principles should also guide the design of new algorithms. To better model users' perceptions, future works should integrate subjective pilot studies in combination with objective metrics to evaluate algorithm performance.

- **Develop user-tailored prediction models:** Although many approaches have been proposed, different AI models and input types may be better suited for different users. Therefore, a more general framework that integrates multiple Artificial Intelligence (AI) models and input modalities could adapt predictions to individual users.
- **Integrate viewport prediction in streaming platforms:** To bridge the gap between the research community and final users, viewport prediction should be integrated into VR streaming platforms. Although this is an active field, as indicated by the efforts in this direction by Meta [109], prediction techniques should first reach sufficient maturity in terms of accuracy, prediction horizons, and generalizability across multiple users.

6. Conclusions

With the diffusion of VR, 360-degree videos are expected to represent a relevant portion of the media content streamed over next-generation networks. Due to the inherent size of this type of video, as well as the limitations of the human visual system, viewport prediction is a key step in the transmission pipeline to reduce the bandwidth requirements without affecting users' QoE.

This paper presents a thorough review of existing AI-based viewport prediction methods emerging in the last ten years. While initial approaches relied on quite simple head movement forecasts, recent methods employ complex DL architectures, which are better suited to learning the complex nature of human exploration behavior. Apart from the type of processing algorithm, methods can be categorized based on the type of input, which spans from human data such as eye or head traces to video content features such as saliency or motion maps.

Despite the rich contributions of the research community, several challenges are still present. Based on this, future research should target the collection of varied and large datasets, as well as increasing the prediction horizon and reducing models' complexity. In addition, attention towards users' perceived QoE should be included in the method design. Solving these issues will foster the application of viewport prediction techniques in real-life scenarios.

Author Contributions: Investigation, M.Z.A.W. and S.B.; writing—original draft preparation, M.Z.A.W. and S.B.; writing—review and editing, S.B. and F.B.; visualization, S.B.; supervision, F.B.; funding acquisition, F.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) Mission 4, Component 2, Investment 1.3, CUP C93C22005250001, partnership on "Telecommunications of the Future" (PE00000001 program "RESTART").

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ARIMA	Autoregressive Integrated Moving Average
AUC-Judd	Area Under the Curve—Judd
BiLSTM	Bidirectional Long Short-Term Memory
CBAM	Convolutional Block Attention Module
CC	Pearson Linear Correlation Coefficient
CNN	Convolutional Neural Network
Conv-LSTM	Convolutional Long Short-Term Memory
DFCA	Dynamic Federated Clustering Algorithm

DHP	Deep Reinforcement Learning-Based Head Movement Prediction
DL	Deep Learning
DoF	Degrees of Freedom
DR	Dead Reckoning
DRL	Deep Reinforcement Learning
ERP	Equirectangular Projection
FN	False Negative
FoV	Field of View
FP	False Positive
FSM	Fused Saliency Map
GBVS	Graph-Based Visual Saliency
GCN	Graph Convolutional Network
GEP	Gaze-Based Field of View Prediction
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
HMD	Head-Mounted Display
IoU	Intersection over Union
KL	Kulback–Leibler Divergence
KNN	K-Nearest Neighbors
LR	Linear Regression
LSTM	Long Short-Term Memory
MIAE	Mean Intersection Angle Error
ML	Machine Learning
MPC	Model Predictive Controller
MSE	Mean Squared Error
NSS	Normalized Scanpath Saliency
PEVQ	Perceptual Evaluation of Video Quality
PoI	Point of Interest
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
RL	Reinforcement learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RoI	Regions of Interest
RR	Ridge Regression
SE-net	Squeeze-and-Excitation Network
SE-Unet	Squeeze-and-Excitation Network and U-Net
SP-ConvGRU	Spherical Convolutional Gated Recursive Unit
SP-GRU	Spherical Gated Recurrent Unit
SSIM	Structural Similarity Index
SVM	Support Vector Machine
SVR	Support Vector Regressor
TN	True Negative
TP	True Positive
TT	Total True
VR	Virtual Reality
VST	Visual Saliency Transformer
VWS-PSNR	Viewport-Weighted PSNR
WLR	Weighted Linear Regression

References

1. Global Industry Analysts Inc. Virtual Reality (VR)-Global Strategic Business Report. 2025. Available online: https://www.researchandmarkets.com/reports/3633908/virtual-reality-vr-global-strategic-business?srsltid=AfmBOookGFSnubhPSmOYKDCSUry2qI5-UrXnjJh24schZn0F6-wX2hjn (accessed on 1 September 2025)
2. Zink, M.; Sitaraman, R.; Nahrstedt, K. Scalable 360° Video Stream Delivery: Challenges, Solutions, and Opportunities. *Proc. IEEE* **2019**, *107*, 639–650. [CrossRef]
3. Nguyen, A.; Yan, Z. Enhancing 360 Video Streaming through Salient Content in Head-Mounted Displays. *Sensors* **2023**, *23*, 4016. [CrossRef] [PubMed]
4. Park, S.; Bhattacharya, A.; Yang, Z.; Das, S.; Samaras, D. Mosaic : Advancing User Quality of Experience in 360-Degree Video Streaming With Machine Learning. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1000–1015. [CrossRef]
5. Islam, R.; Desai, K.; Quarles, J. Cybersickness Prediction from Integrated HMD's Sensors: A Multimodal Deep Fusion Approach using Eye-tracking and Head-tracking Data. In Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Bari, Italy, 4–8 October 2021; pp. 31–40. [CrossRef]
6. Islam, R.; Desai, K.; Quarles, J. Towards Forecasting the Onset of Cybersickness by Fusing Physiological, Head-tracking and Eye-tracking with Multimodal Deep Fusion Network. In Proceedings of the 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Singapore, 17–21 October 2022; pp. 121–130. [CrossRef]
7. Yaqoob, A.; Bi, T.; Muntean, G.M. A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities. *IEEE Commun. Surv. Tutorials* **2020**, *22*, 2801–2838. [CrossRef]
8. Dziubinski, K.; Bandai, M. Local and Global Viewport History Sampling for Improved User Quality of Experience in Viewport-Aware Tile-Based 360-Degree Video Streaming. *IEEE Access* **2024**, *12*, 137455–137471. [CrossRef]
9. Bao, Y.; Wu, H.; Zhang, T.; Ramli, A.A.; Liu, X. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 1161–1170. [CrossRef]
10. Qian, F.; Ji, L.; Han, B.; Gopalakrishnan, V. Optimizing 360 video delivery over cellular networks. In Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, New York, NY, USA, 3–7 October 2016; pp. 1–6. [CrossRef]
11. Qian, F.; Han, B.; Xiao, Q.; Gopalakrishnan, V. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 99–114. [CrossRef]
12. Ban, Y.; Xie, L.; Xu, Z.; Zhang, X.; Guo, Z.; Wang, Y. CUB360: Exploiting Cross-Users Behaviors for Viewport Prediction in 360 Video Adaptive Streaming. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; pp. 1–6. [CrossRef]
13. Baldoni, S.; Poci, O.; Calvagno, G.; Battisti, F. An Ablation Study on 360-Degree Saliency Estimation. In Proceedings of the 2023 International Symposium on Image and Signal Processing and Analysis (ISPA), Rome, Italy, 18–19 September 2023; pp. 1–6. [CrossRef]
14. Aladagli, A.D.; Ekmekcioglu, E.; Jarnikov, D.; Kondoz, A. Predicting head trajectories in 360° virtual reality videos. In Proceedings of the 2017 International Conference on 3D Immersion (IC3D), Brussels, Belgium, 11–12 December 2017; pp. 1–6. [CrossRef]
15. Petrangeli, S.; Simon, G.; Swaminathan, V. Trajectory-Based Viewport Prediction for 360-Degree Virtual Reality Videos. In Proceedings of the 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), Taichung, Taiwan, 10–12 December 2018; pp. 157–160. [CrossRef]
16. Nasrabadi, A.T.; Samiei, A.; Prakash, R. Viewport prediction for 360° videos: A clustering approach. In Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, Istanbul, Turkey, 10–11 June 2020; pp. 34–39. [CrossRef]
17. Li, J.; Wang, Y.; Liu, Y. Meta360: Exploring User-Specific and Robust Viewport Prediction in360-Degree Videos through Bi-Directional LSTM and Meta-Adaptation. In Proceedings of the 2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Sydney, Australia, 16–20 October 2023; pp. 652–661. [CrossRef]
18. Mahmoud, M.; Rizou, S.; Panayides, A.S.; Kantartzis, N.V.; Karagiannidis, G.K.; Lazaridis, P.I.; Zaharis, Z.D. Optimized Tile Quality Selection in Multi-User 360° Video Streaming. *IEEE Open J. Commun. Soc.* **2024**, *5*, 7301–7316. [CrossRef]
19. Feng, W.; Wang, S.; Dai, Y. Adaptive 360-Degree Streaming: Optimizing with Multi-window and Stochastic Viewport Prediction. *IEEE Trans. Mob. Comput.* **2025**, *24*, 5903–5915. [CrossRef]
20. Chao, F.Y.; Ozcinar, C.; Smolic, A. Transformer-based Long-Term Viewport Prediction in 360° Video: Scanpath is All You Need. In Proceedings of the 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), Tampere, Finland, 6–8 October 2021; pp. 1–6. [CrossRef]

21. Wang, H.; Long, Z.; Dong, H.; El Saddik, A. MADRL-Based Rate Adaptation for 360° Video Streaming With Multiviewpoint Prediction. *IEEE Internet Things J.* **2024**, *11*, 26503–26517. [CrossRef]
22. Ao, A.; Park, S. Applying Transformer-Based Computer Vision Models to Adaptive Bitrate Allocation for 360° Live Streaming. In Proceedings of the 2024 IEEE Wireless Communications and Networking Conference (WCNC), Dubai, United Arab Emirates, 21–24 April 2024; pp. 1–6. [CrossRef]
23. Fan, C.L.; Lee, J.; Lo, W.C.; Huang, C.Y.; Chen, K.T.; Hsu, C.H. Fixation Prediction for 360° Video Streaming in Head-Mounted Virtual Reality. In Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video, Taipei, Taiwan, 20–23 June 2017; pp. 67–72. [CrossRef]
24. Nguyen, A.; Yan, Z.; Nahrstedt, K. Your Attention is Unique: Detecting 360-Degree Video Saliency in Head-Mounted Display for Head Movement Prediction. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 22-26 October 2018; pp. 1190–1198. [CrossRef]
25. Li, Y.; Xu, Y.; Xie, S.; Ma, L.; Sun, J. Two-Layer FoV Prediction Model for Viewport Dependent Streaming of 360-Degree Videos. In Proceedings of the International ICST Conference on Communications and Networking in China, Chengdu, China, 23–25 October 2018.
26. Xu, Y.; Dong, Y.; Wu, J.; Sun, Z.; Shi, Z.; Yu, J.; Gao, S. Gaze Prediction in Dynamic 360° Immersive Videos. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5333–5342. [CrossRef]
27. Fan, C.L.; Yen, S.C.; Huang, C.Y.; Hsu, C.H. Optimizing Fixation Prediction Using Recurrent Neural Networks for 360° Video Streaming in Head-Mounted Virtual Reality. *IEEE Trans. Multimed.* **2020**, *22*, 744–759. [CrossRef]
28. Feng, X.; Liu, Y.; Wei, S. LiveDeep: Online Viewport Prediction for Live Virtual Reality Streaming Using Lifelong Deep Learning. In Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Atlanta, GA, USA, 22–26 March 2020; pp. 800–808. [CrossRef]
29. Rondón, M.F.R.; Sassatelli, L.; Aparicio-Pardo, R.; Precioso, F. TRACK: A New Method From a Re-Examination of Deep Architectures for Head Motion Prediction in 360° Videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5681–5699. [CrossRef]
30. Manfredi, G.; Racanelli, V.A.; De Cicco, L.; Mascolo, S. LSTM-based Viewport Prediction for Immersive Video Systems. In Proceedings of the 2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet), Island of Ponza, Italy, 13–15 June 2023; pp. 49–52. [CrossRef]
31. Wang, M.; Peng, S.; Chen, X.; Zhao, Y.; Xu, M.; Xu, C. CoLive: An Edge-Assisted Online Learning Framework for Viewport Prediction in 360° Live Streaming. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6. [CrossRef]
32. Dong, P.; Shen, R.; Xie, X.; Li, Y.; Zuo, Y.; Zhang, L. Predicting Long-Term Field of View in 360-Degree Video Streaming. *IEEE Netw.* **2023**, *37*, 26–33. [CrossRef]
33. Wang, M.; Chen, X.; Yang, X.; Peng, S.; Zhao, Y.; Xu, M.; Xu, C. CoLive: Edge-Assisted Clustered Learning Framework for Viewport Prediction in 360° Live Streaming. *IEEE Trans. Multimed.* **2024**, *26*, 5078–5091. [CrossRef]
34. Zhang, L.; Zhou, H.; Shen, L.; Liu, J.; Cui, L. Towards Attention-Aware Interactive 360-Degree Video Streaming on Smartphones. *IEEE Netw.* **2025**, *39*, 147–156. [CrossRef]
35. Li, J.; Han, L.; Zhang, C.; Li, Q.; Liu, Z. Spherical Convolution Empowered Viewport Prediction in 360 Video Multicast with Limited FoV Feedback. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 3. [CrossRef]
36. Setayesh, M.; Wong, V.W.S. Viewport Prediction, Bitrate Selection, and Beamforming Design for THz-Enabled 360° Video Streaming. *IEEE Trans. Wirel. Commun.* **2025**, *24*, 1849–1865. [CrossRef]
37. Zhang, Z.; Du, H.; Huang, S.; Zhang, W.; Zheng, Q. VRFormer: 360-Degree Video Streaming with FoV Combined Prediction and Super resolution. In Proceedings of the 2022 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking (ISPA/BDCloud/SocialCom/SustainCom), Melbourne, Australia, 17–19 December 2022; pp. 531–538. [CrossRef]
38. Gao, B.; Sheng, D.; Zhang, L.; Qi, Q.; He, B.; Zhuang, Z.; Wang, J. STAR-VP: Improving Long-term Viewport Prediction in 360° Videos via Space-aligned and Time-varying Fusion. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, Australia, 28 October–1 November 2024; pp. 5556–5565. [CrossRef]
39. Guo, Y.; Xu, M.; Jiang, L.; Deng, X.; Zhou, J.; Chen, G.; Sigal, L. Proposal With Alignment: A Bi-Directional Transformer for 360° Video Viewport Proposal. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 11423–11437. [CrossRef]
40. Xu, X.; Tan, X.; Wang, S.; Liu, Z.; Zheng, Q. Multi-Features Fusion based Viewport Prediction with GNN for 360-Degree Video Streaming . In Proceedings of the 2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom), Kyoto, Japan, 26–28 June 2023; pp. 57–64. [CrossRef]
41. Xu, M.; Song, Y.; Wang, J.; Qiao, M.; Huo, L.; Wang, Z. Predicting Head Movement in Panoramic Video: A Deep Reinforcement Learning Approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2693–2708. [CrossRef]

42. Wu, C.; Tan, Z.; Wang, Z.; Yang, S. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming. In Proceedings of the 8th ACM on Multimedia Systems Conference, Taipei, Taiwan, 20–23 June 2017; pp. 193–198. [CrossRef]
43. Bao, Y.; Zhang, T.; Pande, A.; Wu, H.; Liu, X. Motion-Prediction-Based Multicast for 360-Degree Video Transmissions. In Proceedings of the 2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), San Diego, CA, USA, 12–14 June 2017; pp. 1–9. [CrossRef]
44. Markley, L.; Cheng, Y.; Crassidis, J.; Oshman, Y. Averaging Quaternions. *J. Guid. Control. Dyn.* **2007**, *30*, 1193–1196. [CrossRef]
45. Nasrabadi, A.T.; Samiei, A.; Mahzari, A.; McMahan, R.P.; Prakash, R.; Farias, M.C.Q.; Carvalho, M.M. A taxonomy and dataset for 360° videos. In Proceedings of the 10th ACM Multimedia Systems Conference, Amherst, MA, USA, 18–21 June 2019; pp. 273–278. [CrossRef]
46. Lo, W.C.; Fan, C.L.; Lee, J.; Huang, C.Y.; Chen, K.T.; Hsu, C.H. 360° Video Viewing Dataset in Head-Mounted Virtual Reality. In Proceedings of the 8th ACM on Multimedia Systems Conference, Taipei, Taiwan, 20–23 June 2017; pp. 211–216. [CrossRef]
47. Li, C.; Xu, M.; Du, X.; Wang, Z. Bridge the Gap Between VQA and Human Behavior on Omnidirectional Video: A Large-Scale Dataset and a Deep Learning Model. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 932–940. [CrossRef]
48. Developers, F. FFmpeg Multimedia Framework. Available online: https://ffmpeg.org/ (accessed on 29 July 2025).
49. Guo, C.; Cui, Y.; Liu, Z. Optimal Multicast of Tiled 360 VR Video in OFDMA Systems. *IEEE Commun. Lett.* **2018**, *22*, 2563–2566. [CrossRef]
50. Mahmoud, M.; Valiandi, I.; Panayides, A.S.; Rizou, S.; Lazaridis, P.I.; Karagiannidis, G.K.; Kantartzis, N.V.; Zaharis, Z.D. Versatile Video Coding Performance Evaluation for Tiled 360 deg Videos. In Proceedings of the European Wireless 2023; 28th European Wireless Conference, Rome, Italy, 2–4 October 2023; pp. 191–196.
51. David, E.J.; Gutiérrez, J.; Coutrot, A.; Da Silva, M.P.; Callet, P.L. A dataset of head and eye movements for 360° videos. In Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, The Netherlands, 12–15 June 2018; pp. 432–437. [CrossRef]
52. Hou, X.; Dey, S.; Zhang, J.; Budagavi, M. Predictive Adaptive Streaming to Enable Mobile 360-Degree and VR Experiences. *IEEE Trans. Multimed.* **2021**, *23*, 716–731. [CrossRef]
53. Zhang, Y.; Zhao, P.; Bian, K.; Liu, Y.; Song, L.; Li, X. DRL360: 360-degree Video Streaming with Deep Reinforcement Learning. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1252–1260. [CrossRef]
54. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–12 August 2017; Volume 70, pp. 1126–1135.
55. Lu, Y.; Zhu, Y.; Wang, Z. Personalized 360-Degree Video Streaming: A Meta-Learning Approach. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 3143–3151. [CrossRef]
56. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
57. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
58. Itti, L.; Koch, C.; Niebur, E. A Model of Saliency-based Visual Attention for Rapid Scene Analysis. *Pattern Anal. Mach. Intell. IEEE Trans.* **1998**, *20*, 1254 – 1259. [CrossRef]
59. Lou, J.; Lin, H.; Marshall, D.; Saupe, D.; Liu, H. TranSalNet: Towards perceptually relevant visual saliency prediction. *Neurocomputing* **2022**, *494*, 455–467. [CrossRef]
60. Liu, N.; Zhang, N.; Wan, K.; Shao, L.; Han, J. Visual Saliency Transformer. *arXiv* **2021**, arXiv:2104.12099. . [CrossRef]
61. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–80. [CrossRef]
62. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; Volume 2, pp. 674–679.
63. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
64. Cornia, M.; Baraldi, L.; Serra, G.; Cucchiara, R. A deep multi-level network for saliency prediction. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 3488–3493. [CrossRef]
65. Mavlankar, A.; Girod, B., Video Streaming with Interactive Pan/Tilt/Zoom. In *High-Quality Visual Experience: Creation, Processing and Interactivity of High-Resolution and High-Dimensional Video Signals*; Mrak, M., Grgic, M., Kunt, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 431–455. [CrossRef]
66. El-Ganainy, T.; Hefeeda, M. Streaming virtual reality content. *arXiv* **2016**, arXiv:1612.08350. [CrossRef]

67. Corbillon, X.; Simon, G.; Devlic, A.; Chakareski, J. Viewport-adaptive navigable 360-degree video delivery. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7. [CrossRef]
68. Yu, M.; Lakshman, H.; Girod, B. A Framework to Evaluate Omnidirectional Video Coding Schemes. In Proceedings of the 2015 IEEE International Symposium on Mixed and Augmented Reality, Fukuoka, Japan, 29 September–3 October 2015; pp. 31–36. [CrossRef]
69. Pan, J.; Sayrol, E.; Giro-i Nieto, X.; McGuinness, K.; O'Connor, N.E. Shallow and Deep Convolutional Networks for Saliency Prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
70. Huang, X.; Shen, C.; Boix, X.; Zhao, Q. SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 262–270. [CrossRef]
71. Xie, L.; Xu, Z.; Ban, Y.; Zhang, X.; Guo, Z. 360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 315–323. [CrossRef]
72. De Abreu, A.; Ozcinar, C.; Smolic, A. Look around you: Saliency maps for omnidirectional images in VR applications. In Proceedings of the 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX), Erfurt, Germany, 31 May–2 June 2017; pp. 1–6. [CrossRef]
73. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-ResNet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284. [CrossRef]
74. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2462–2470.
75. Feng, X.; Swaminathan, V.; Wei, S. Viewport Prediction for Live 360-Degree Mobile Video Streaming Using User-Content Hybrid Motion Tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 43. [CrossRef]
76. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.k.; Woo, W.c. Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. In Proceedings of the 29th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 1; pp. 802–810.
77. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
78. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [CrossRef]
79. Nguyen, A.; Yan, Z. A saliency dataset for 360-degree videos. In Proceedings of the 10th ACM Multimedia Systems Conference, Amherst, MA, USA, 18–21 June 2019; pp. 279–284. [CrossRef]
80. Zhang, L.; Suo, Y.; Wu, X.; Wang, F.; Chen, Y.; Cui, L.; Liu, J.; Ming, Z. TBRA: Tiling and Bitrate Adaptation for Mobile 360-Degree Video Streaming. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, 20–24 October 2021; pp. 4007–4015. [CrossRef]
81. Dasari, M.; Bhattacharya, A.; Vargas, S.; Sahu, P.; Balasubramanian, A.; Das, S.R. Streaming 360-Degree Videos Using Super-Resolution. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 1977–1986. [CrossRef]
82. Peng, S.; Hu, J.; Li, Z.; Xiao, H.; Yang, S.; Xu, C. Spherical Convolution-based Saliency Detection for FoV Prediction in 360-degree Video Streaming. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; pp. 162–167. [CrossRef]
83. Coors, B.; Condurache, A.P.; Geiger, A. SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
84. Zhang, Z.; Xu, Y.; Yu, J.; Gao, S. Saliency Detection in 360° Videos. In Proceedings of the Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, 8–14 September 2018; Proceedings, Part VII; Springer: Berlin/Heidelberg, Germany 2018; pp. 504–520. [CrossRef]
85. Cho, K.; Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014. [CrossRef]
86. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521. [CrossRef]

87. Chao, F.Y.; Zhang, L.; Hamidouche, W.; Deforges, O. Salgan360: Visual Saliency Prediction On 360 Degree Images With Generative Adversarial Networks. In Proceedings of the 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), San Diego, CA, USA, 23–27 July 2018; pp. 1–4. [CrossRef]
88. Jiang, L.; Xu, M.; Liu, T.; Qiao, M.; Wang, Z. DeepVS: A Deep Learning Based Video Saliency Prediction Approach. In Proceedings of the Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, 8–14 September 2018; Proceedings, Part XIV; Spring: Berlin/Heidelberg, Germany, 2018; pp. 625–642. [CrossRef]
89. Che, Z.; Borji, A.; Zhai, G.; Min, X.; Guo, G.; Le Callet, P. How is Gaze Influenced by Image Transformations? Dataset and Model. *Trans. Img. Proc.* **2020**, *29*, 2287–2300. [CrossRef] [PubMed]
90. Setayesh, M.; Wong, V.W. A Content-based Viewport Prediction Framework for 360° Video Using Personalized Federated Learning and Fusion Techniques. In Proceedings of the 2023 IEEE International Conference on Multimedia and Expo (ICME), Brisbane, Australia, 10–14 July 2023; pp. 654–659. [CrossRef]
91. Yun, H.; Lee, S.; Kim, G. Panoramic Vision Transformer for Saliency Detection in 360° Videos. *arXiv* **2022**, arXiv:2209.08956. [CrossRef]
92. Liu, X.; Deng, Y.; Han, C.; Renzo, M.D. Learning-Based Prediction, Rendering and Transmission for Interactive Virtual Reality in RIS-Assisted Terahertz Networks. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 710–724. [CrossRef]
93. Yaqoob, A.; Muntean, G.M. A Combined Field-of-View Prediction-Assisted Viewport Adaptive Delivery Scheme for 360° Videos. *IEEE Trans. Broadcast.* **2021**, *67*, 746–760. [CrossRef]
94. Huang, R.; Wong, V.W.; Schober, R. Rate-Splitting for Intelligent Reflecting Surface-Aided Multiuser VR Streaming. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1516–1535. [CrossRef]
95. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [CrossRef]
96. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part IV 14; Springer: Cham, Switzerland, 2016; pp. 630–645.
97. Pang, H.; Zhang, C.; Wang, F.; Liu, J.; Sun, L. Towards Low Latency Multi-viewpoint 360° Interactive Video: A Multimodal Deep Reinforcement Learning Approach. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 991–999. [CrossRef]
98. Li, C.; Zhang, W.; Liu, Y.; Wang, Y. Very Long Term Field of View Prediction for 360-Degree Video Streaming. In Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 28–30 March 2019; pp. 297–302. [CrossRef]
99. Chen, J.; Hu, M.; Luo, Z.; Wang, Z.; Wu, D. SR360: Boosting 360-degree video streaming with super-resolution. In Proceedings of the NOSSDAV '20: Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, Istanbul, Turkey, 10–11 June 2020; pp. 1–6. [CrossRef]
100. Zhang, Z.; Chen, Y.; Zhang, W.; Yan, C.; Zheng, Q.; Wang, Q.; Chen, W. Tile Classification Based Viewport Prediction with Multi-modal Fusion Transformer. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023; MM '23, pp. 3560–3568. [CrossRef]
101. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning Spatio-Temporal Transformer for Visual Tracking. *arXiv* **2021**, arXiv:2103.17154. [CrossRef]
102. Meinhardt, T.; Kirillov, A.; Leal-Taixé, L.; Feichtenhofer, C. TrackFormer: Multi-Object Tracking with Transformers. *arXiv* **2021**, arXiv:2101.02702. [CrossRef]
103. Dahou Djilali, Y.A.; Tliba, M.; McGuinness, K.; O'Connor, N. ATSal: An Attention Based Architecture for Saliency Prediction in 360 Videos. *arXiv* **2020**, arXiv:2011.10600. [CrossRef]
104. Vo, C.H.; Chiang, J.C.; Le, D.H.; Nguyen, T.T.; Pham, T.V. Saliency Prediction for 360-degree Video. In Proceedings of the 2020 5th International Conference on Green Technology and Sustainable Development (GTSD), Ho Chi Minh City, Vietnam, 27–28 November 2020; pp. 442–448. [CrossRef]
105. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; NIPS'16; pp. 3844–3852.
106. Chopra, L.; Chakraborty, S.; Mondal, A.; Chakraborty, S. PARIMA: Viewport Adaptive 360-Degree Video Streaming. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; ACM: New York, NY, USA 2021; WWW '21. [CrossRef]
107. Hu, H.N.; Lin, Y.C.; Liu, M.Y.; Cheng, H.T.; Chang, Y.J.; Sun, M. Deep 360 Pilot: Learning a Deep Agent for Piloting through 360° Sports Videos. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1396–1405. [CrossRef]

108. Chen, P.W.; Yang, T.S.; Huang, G.L.; Huang, C.W.; Chao, Y.C.; Lu, C.H.; Wu, P.Y. Viewing Bias Matters in 360° Videos Visual Saliency Prediction. *IEEE Access* **2023**, *11*, 46084–46094. [CrossRef]
109. Chen, S.; Kuzyakov, E.; Peng, R. Enhancing High-Resolution 360 Streaming with View Prediction. 2017. Available online: https://engineering.fb.com/2017/04/19/virtual-reality/enhancing-high-resolution-360-streaming-with-view-prediction/ (accessed on 31 August 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Electronics Editorial Office
E-mail: electronics@mdpi.com
www.mdpi.com/journal/electronics



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-8790-3