

Published in Journals: AI, Buildings, Computers, Drones, Entropy and Symmetry

Topic Reprint

Applications of Machine Learning in Large-Scale Optimization and High-Dimensional Learning

Edited by
Jeng-Shyang Pan, Junzo Watada, Vaclav Snasel and Pei Hu

mdpi.com/topics



Applications of Machine Learning in Large-Scale Optimization and High-Dimensional Learning

Applications of Machine Learning in Large-Scale Optimization and High-Dimensional Learning

Topic Editors

Jeng-Shyang Pan

Junzo Watada

Vaclav Snasel

Pei Hu



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

Topic Editors

Jeng-Shyang Pan

College of Artificial

Intelligence

Nanjing University of

Information Science and

Technology

Nanjing

China

Pei Hu

School of Computer and

Software

Nanyang Institute of

Technology

Nanyang

China

Junzo Watada

Graduate School of

Information, Production and

Systems

Waseda University

Tokyo

Japan

Vaclav Snasel

VSB

Technical University of

Ostrava

Ostrava

Czech Republic

Editorial Office

MDPI AG

Grosspeteranlage 5

4052 Basel, Switzerland

This is a reprint of the Topic, published open access by the journals *AI* (ISSN 2673-2688), *Buildings* (ISSN 2075-5309), *Computers* (ISSN 2073-431X), *Drones* (ISSN 2504-446X), *Entropy* (ISSN 1099-4300) and *Symmetry* (ISSN 2073-8994), freely accessible at: <https://www.mdpi.com/topics/UK78275010>.

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. <i>Journal Name</i> Year , Volume Number, Page Range.
--

ISBN 978-3-7258-9047-7 (Hbk)

ISBN 978-3-7258-9048-4 (PDF)

<https://doi.org/10.3390/books978-3-7258-9048-4>

© 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

About the Editors	vii
-----------------------------	-----

Zongming Liu, Yue Zhu and Zihao Wang A Recognition Method for Architectural Decorative Motifs in Guanzhong Traditional Vernacular Dwellings for the Digital Documentation of Architectural Heritage Reprinted from: <i>Buildings</i> 2026 , 16, 3003, https://doi.org/10.3390/buildings16153003	1
--	---

Chao Zhang, Yufei Zhang, Feng Wang, Xiaoxu Su, Zhijie Dong and Linlin Zuo A Testability Strategy Optimization Method Under Multi-Valued Dependency Condition Based on Deep Reinforcement Learning Reprinted from: <i>Entropy</i> 2026 , 28, 733, https://doi.org/10.3390/e28070733	20
---	----

Meilin Liu, Wenping Zheng and Shuxia Yuan DDEC: Dual Dependency-Enhanced Contrastive Learning for Sparse Hypergraph Node Classification Reprinted from: <i>Entropy</i> 2026 , 28, 729, https://doi.org/10.3390/e28070729	54
---	----

Mohsen Golafrouz, Houshyar Asadi, Mohammad Reza Chalak Qazani, Anwar Hosen, Zoran Najdovski, Lei Wei, et al. Optimized Deep Learning Framework for Emotion Recognition Using Multimodal Physiological Signals and Temporal Convolutional Networks Reprinted from: <i>Computers</i> 2026 , 15, 381, https://doi.org/10.3390/computers15060381	76
---	----

Jiale Li, Jicong Duan, Changbin Shao and Hualong Yu PHASE: Progressive Hierarchical Adaptation for Sample-Efficient Rebalancing in Long-Tail Classification Reprinted from: <i>Symmetry</i> 2025 , 17, 2040, https://doi.org/10.3390/sym17122040	93
---	----

Chen Liu, Qihong Feng, Wensheng Zhou, Chi Zhang and Xianmin Zhang Flow Field Evaluation Method of High Water-Cut Reservoirs Based on K-Means Clustering Algorithm Reprinted from: <i>Symmetry</i> 2025 , 17, 901, https://doi.org/10.3390/sym17060901	110
--	-----

Fei Yu, Guiju Wu, Yufei Xi, Fan Wang, Jiapeli Wang, Rui Zhang and Qinghong Long Inversion of Gravity Anomalies Based on U-Net Network Reprinted from: <i>Symmetry</i> 2025 , 17, 523, https://doi.org/10.3390/sym17040523	128
---	-----

Yuhao Liu, Junjie Hou, Ping Wei, Jie Jin and Renjie Zhang Research and Application of ROM Based on Res-PINNs Neural Network in Fluid System Reprinted from: <i>Symmetry</i> 2025 , 17, 163, https://doi.org/10.3390/sym17020163	151
---	-----

Bashar Khawaldeh, Antonio M. Mora and Hossam Faris Optimizing Steering Angle Prediction in Self-Driving Vehicles Using Evolutionary Convolutional Neural Networks Reprinted from: <i>AI</i> 2024 , 5, 105, https://doi.org/10.3390/ai5040105	172
---	-----

Gabriel Provencher Langlois, Jatan Buch and Jérôme Darbon Efficient First-Order Algorithms for Large-Scale, Non-Smooth Maximum Entropy Models with Application to Wildfire Science Reprinted from: <i>Entropy</i> 2024 , 26, 691, https://doi.org/10.3390/e26080691	195
--	-----

Hongbin Sun, Qiuchen Shen, Hongchang Ke, Zhenyu Duan and Xi Tang Power Transmission Lines Foreign Object Intrusion Detection Method for Drone Aerial Images Based on Improved YOLOv8 Network Reprinted from: <i>Drones</i> 2024 , 8, 346, https://doi.org/10.3390/drones8080346	219
Ange Gabriel Belinga, Cédric Stéphane Tekouabou Koumetio, Mohammed El Haziti and Mohammed El Hassouni Knowledge Distillation in Image Classification: The Impact of Datasets Reprinted from: <i>Computers</i> 2024 , 13, 184, https://doi.org/10.3390/computers13080184	245

About the Editors

Jeng-Shyang Pan

Jeng-Shyang Pan is a Taiwanese scholar, a fellow of the Institution of Engineering and Technology (IET), a professor and doctoral supervisor, and a national specially appointed expert. He has long been deeply involved in research fields such as intelligent computing and multimedia signal processing, and has continuously promoted academic exchanges and scientific research collaborations between the two sides of the Taiwan Strait. Early in his career, he worked at Kaohsiung University of Applied Sciences in Taiwan, holding administrative positions such as department head and vice dean of the college. Since 1997, he has maintained long-term collaborations with universities in mainland China, successively serving as a specially appointed professor at the Harbin Institute of Technology, dean and assistant president of Fujian University of Technology, and professor at Shandong University of Science and Technology. He currently serves at the School of Artificial Intelligence, Nanjing University of Information Science and Technology, and also holds adjunct, visiting, and specially appointed teaching positions at several other universities in China. His research achievements are substantial, covering multiple areas including intelligent computing, image processing, and information security. He has published over 700 academic papers, with over 300 indexed by SCI, being cited over 17,000 times and achieving an H-index of 63. He has also published over 10 monographs, spearheaded the founding of an SCI journal, led 38 research projects, and obtained over 50 invention patents. He has received numerous prestigious honors, including the Second Prize of Natural Science from the Ministry of Education and the title of Outstanding People's Teacher of Fujian Province. He has been recognized as one of the world's top scientists in the field of computer science and has made significant achievements and outstanding contributions to discipline development and cross-strait research cooperation.

Junzo Watada

Junzo Watada is an Emeritus Professor at Waseda University, Japan, and currently a full-time professor at Universiti Petronas, Malaysia. He received his Bachelor's, Master's, and Doctor of Engineering degrees in electrical engineering from Osaka City University and Osaka Prefecture University, respectively. He previously worked as a systems engineer at Fujitsu and has long been a dedicated teacher and researcher at Waseda University. He is a Senior Life Member of IEEE, a fellow of several international societies, and has served as president of numerous international societies and held important academic positions. His main research areas include soft computing, fuzzy systems, big data analytics, data mining, image processing, management engineering, and financial decision-making systems. He has a prolific academic record, having published over 600 papers and authored or co-authored over 40 books and chapters. He has served as Editor-in-Chief and Editorial Board Member of several renowned international journals and has received numerous international academic medals and lifetime achievement awards. He has a profound international influence in the fields of soft computing and intelligent information processing.

Vaclav Snasel

Vaclav Snasel is a full-time professor in the Department of Computer Science at the Technical University of Ostrava (VSB-TUO) in the Czech Republic, a Senior Member of IEEE, and has over 30 years of experience in industry, academia, and research. He holds a Master of Science degree from Palacký University and a PhD in algebra and geometry from Masaryk University. He had nearly eight years of experience in corporate R&D before joining the faculty at VSB-TUO in 2001. He was promoted to full professor in 2006 and has served as the vice dean of research at the Faculty of Electrical Engineering and Computer Science since 2003. He also holds positions as a visiting scientist at the Institute of Computer Science of the Czech Academy of Sciences and the director of MIR Labs Europe. His research spans multiple fields, including artificial intelligence, data mining, neural networks, metaheuristic optimization, social networks, formal concept analysis, information retrieval, semantic web, knowledge management, data compression, bio-inspired computing, and Industry 4.0. He has achieved fruitful academic results, publishing over 830 papers with a peak H-index of 47 and over 11,700 citations. He has led numerous international research projects, mentored doctoral students from various countries, served as Editor-in-Chief and Editorial Board member of several international journals, and delivered over 17 keynote speeches and lectures at global academic conferences, demonstrating broad international influence in the fields of intelligent computing and data science.

Pei Hu

Pei Hu, PhD, is a lecturer and member of the Cloud Computing Technology Teaching and Research Section at the School of Computer Science and Software Engineering, Nanyang Institute of Technology. His main research areas are evolutionary computing, feature selection, data mining, and artificial intelligence. He is a member of the Granular Computing and Its Applications Professional Committee of the Chinese Association for Artificial Intelligence (CAA). He has published over 30 SCI papers, with a Google Scholar H-index of 12 and over 900 citations; has published one monograph; and holds three authorized invention patents. In academic service, he served as the publishing chair for the IIH-MSP 2022 and ICGEC 2023 international conferences and as a Guest Editor for a Special Issue of the international journal *Entropy*. He teaches courses such as 'Fundamentals of Object-Oriented Programming' and 'Fundamentals of Big Data'. He has received awards such as Advanced Worker and Outstanding Teacher for Off-Duty Practical Training from the Nanyang Institute of Technology; in teaching, he was awarded Best Instructor in the National Information Technology Application Level Competition and guided students to win second prize in the provincial competition of the National Undergraduate Software Testing Competition.

Article

A Recognition Method for Architectural Decorative Motifs in Guanzhong Traditional Vernacular Dwellings for the Digital Documentation of Architectural Heritage

Zongming Liu ¹, Yue Zhu ^{1,*} and Zihao Wang ²

¹ College of Art and Design, Shaanxi University of Science and Technology, Xi'an 710021, China; liuzm@sust.edu.cn

² College of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China; 250511047@sust.edu.cn

* Correspondence: bs231011011@sust.edu.cn; Tel.: +86-15251766907

Abstract

Guanzhong traditional vernacular dwellings are an important component of traditional architectural heritage. Their architectural decorative motifs are often attached to components such as screen walls, gable end piers, doors and windows, door pillow stones, column bases, tile ends, and roof ridges, making them important visual objects for image based documentation, typological organization, and conservation management of architectural heritage. Since the relevant images are mainly obtained through field photography, field images often contain multiple sources of interference, including weathering, spalling, occlusion, contamination, illumination variation, wall joints, brick joints, shadows, component edges, and damaged textures. These factors lead to blurred motif boundaries, strong background interference, and a high risk of misrecognition. Existing motif recognition methods mainly focus on clear samples and pay insufficient attention to candidate region generation, invalid texture filtering, and the exclusion of low confidence results in complex field scenarios. To address these problems, this study proposes a recognition method for architectural decorative motifs in complex scenarios. Based on field collected images of Guanzhong traditional vernacular dwellings, the method takes architectural decorative motifs and non-motif candidate regions as training objects. SAM is first used to generate potential motif candidate regions. Spatial-prior filtering based on candidate size, aspect ratio, and SAM stability is then introduced to reduce invalid region proposals, while semantic aggregation is used to retain large candidate regions supported by high-confidence local motif evidence. Swin Transformer is further adopted as the classification backbone to extract local texture features and overall structural relationships from candidate regions. A maximum confidence based rejection mechanism is also introduced to enhance the model's ability to exclude low confidence predictions and regions without motifs. Component-level test-set evaluation and expert-verified complete-field-image evaluation show that the proposed method can reduce the interference of complex backgrounds and invalid textures, and improve the recognition stability of architectural decorative regions in field images. This study provides technical support for image organization, digital documentation, typological indexing, and conservation management of architectural decorative motifs in traditional vernacular dwellings.

Keywords: image based documentation of architectural heritage; Guanzhong traditional vernacular dwellings; architectural decorative motifs; complex field images; candidate region segmentation; Swin Transformer; rejection mechanism

1. Introduction

Architectural decorative motifs are visual objects with clear recognition value in the image archives of traditional vernacular dwellings. Compared with general building façades or spatial layouts, decorative motifs are usually embedded in specific architectural components at a relatively small scale [1]. They reveal craft lines, material textures, and compositional relationships, while also recording the component system and visual order of traditional dwellings. Guanzhong traditional vernacular dwellings preserve relatively complete local building experience. Their architectural decoration is commonly attached to parts such as gatehouses, screen walls, gable end piers, doors and windows, door pillow stones, column bases, tile ends, and roof ridges, forming a motif image system characterized by spatial attachment [2]. These motifs are not isolated patterns. Together with architectural components, courtyard entrances, wall boundaries, and roof nodes, they constitute architectural surfaces that can be observed, recorded, and classified. Therefore, image recognition of architectural decorative motifs in Guanzhong traditional vernacular dwellings can improve the efficiency of organizing field image materials and provide basic data support for the digital documentation, typological statistics, and image retrieval of decorative motifs in traditional dwellings [3,4].

From an international perspective, digital documentation has become an important technical basis for cultural heritage conservation and management. Remondino and Rizzi emphasized that reality-based documentation can provide reliable visual and spatial records for heritage sites, while Remondino further discussed the role of photogrammetry and 3D scanning in heritage recording and modeling [5,6]. In the field of architectural heritage management, H-BIM studies have also shown that structured digital information is useful for documenting, organizing, and reusing heritage data [7]. These studies indicate that image-based recognition is not only a classification task, but also a supporting technique for heritage information organization and digital documentation.

During preliminary field investigations, the research team found that image acquisition was often affected by shooting distance, lighting conditions, component height, and viewing angle [8]. These factors resulted in cluttered backgrounds, inconsistent scales, and unclear boundaries. At the same time, as the components of traditional dwellings have long been exposed to the natural environment, weathering, spalling, occlusion, and contamination are common, further weakening the lines, contours, and categorical features of the motifs [9]. These field conditions create difficulties in subsequent image organization and make automated motif recognition necessary for improving efficiency and consistency.

In recent years, deep learning methods have been applied to cultural heritage image analysis, historic architectural component recognition, and the detection of surface deterioration in buildings [10]. Fiorucci et al. reviewed machine learning applications in cultural heritage and pointed out that classification, retrieval, and visual analysis have become important directions in computational heritage studies [11]. For architectural heritage images, Llamas et al. applied deep learning techniques to classify architectural heritage elements and demonstrated that CNN-based image classification can support the organization of architectural image archives [12]. More recently, Croce et al. explored the use of artificial intelligence for architectural heritage classification in an H-BIM context, showing that visual classification can assist semi-automatic heritage reconstruction and information modeling [13]. Existing studies have generally followed two technical pathways. One relies mainly on classification networks such as CNN [14], ViT [15], and Swin Transformer [16], using whole image input or manually cropped images for category prediction. The other

uses semantic segmentation, instance segmentation, or visual foundation models to extract possible target regions from complex images. These methods provide useful technical references for architectural heritage image recognition. However, when whole field images are directly fed into a classification network, the model is often affected by large wall surfaces, component edges, shadows, and damaged textures. This can lead to unstable feature extraction, increased computational redundancy, and a higher rate of misrecognition. It is therefore necessary to develop a recognition method for architectural decorative motifs in complex scene images.

To address these issues, this study constructs a sequential recognition workflow for architectural decorative motifs in Guanzhong traditional vernacular dwellings. The workflow combines non-motif candidate training, SAM-based candidate generation, spatial-prior filtering, semantic aggregation, Swin Transformer classification, and confidence-based rejection to improve recognition stability in complex field images.

The main contributions of this study are as follows:

1. A dataset of architectural decorative motif images from Guanzhong traditional vernacular dwellings is constructed. Non-motif regions such as wall joints, brick joints, stains, shadows, and weathered wood textures are incorporated into the sample system, in order to reflect the practical problem that motifs and background textures are often mixed in field images of architectural heritage.
2. This study does not aim to propose a new general-purpose segmentation or detection model. Its contribution lies in adapting foundation-model candidate generation to complex architectural heritage field images through a dual-branch workflow that combines macroscopic candidate-region anchoring with fine-grained motif classification. The workflow integrates spatial-prior filtering, semantic aggregation, long-tailed performance evaluation, and confidence-based rejection to improve recognition stability under complex field backgrounds.
3. Component-level grouped test-set evaluation, module configuration analysis, and complete-field-image expert verification are conducted to verify the effectiveness of the proposed method in organizing architectural decorative motifs in traditional vernacular dwellings.

2. Research Object and Sample Construction

2.1. Research Object and Image Sources

From January to March 2026, the research team conducted field investigations of traditional dwellings in Shaanxi Province. The study area covered five cities in the Guanzhong region, namely Xi'an, Baoji, Xianyang, Weinan, and Tongchuan (Figure 1). Traditional dwellings in this region are widely distributed, and the preservation conditions of their decorative components vary considerably. These characteristics make the region suitable for reflecting the difficulties of motif recognition under complex conditions.

The research images consist of three levels: complete building photographs, architectural component photographs, and cropped local motif images. Complete images were used to record the dwelling environment and the spatial context of architectural components. Component images were used to identify the parts to which the motifs were attached. Cropped local images served as the basic sample units for model training, validation, and testing. The samples were classified according to five fields, namely city, district or county, dwelling, component name, and specific motif. This classification structure ensures source traceability and facilitates the subsequent indexing, retrieval, and archival management of decorative images from traditional vernacular dwellings. The samples were mainly collected from decorative components such as screen walls, gable end piers, doors and windows, door pillow stones, column bases, tile ends, and roof ridges. The

component types include entrance decorations that can be clearly identified at close range, as well as small scale local motifs that are more strongly affected by viewing angle and preservation condition.

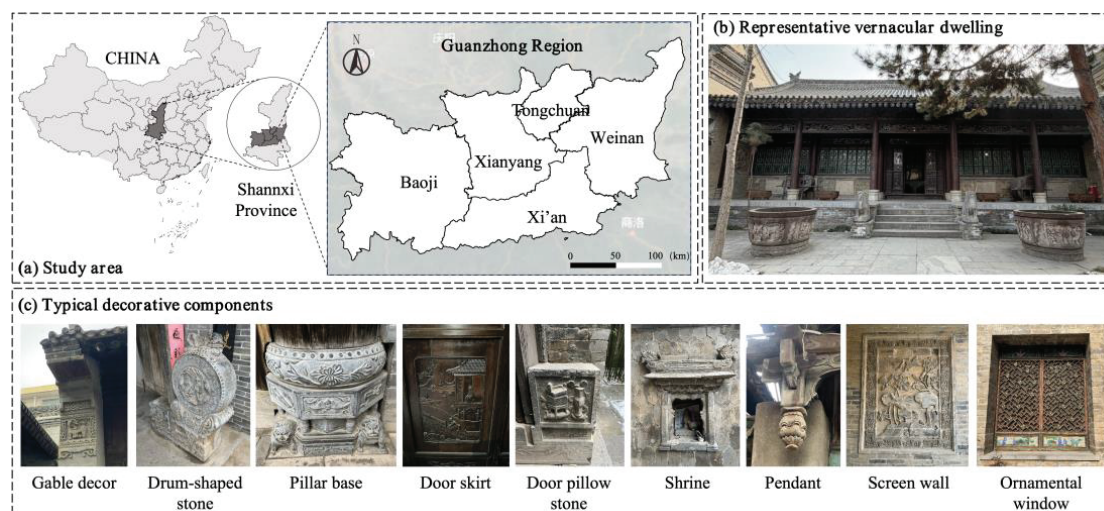


Figure 1. Study area, dwelling locations, and representative decorative components in Guanzhong traditional vernacular dwellings.

After screening, cropping, merging, and manual annotation, the dataset contained 47 traditional dwellings and 4706 annotated image samples. These samples were organized into 164 annotated categories, including 163 architectural decorative motif categories and one non-motif background category. To clarify the long-tailed distribution of the dataset, Table 1 reports the number of samples in each first-level category, the number of second-level categories, and the minimum, maximum, mean, and median number of samples per category. Non-motif candidates mainly include wall joints, brick joints, stains, shadows, weathered wood textures, damaged stone surfaces, and blank wall surfaces. They were introduced as negative samples for rejection training to reduce the probability that invalid textures would be misidentified as decorative motifs. Rare categories were retained to preserve the diversity of Guanzhong vernacular decorative motifs.

Table 1. Distribution of first-level motif categories and non-motif candidates.

First-Level Category	Samples	Second-Level Classes	Min	Max	Mean	Median
Plant motifs	1917	52	1	360	36.87	5
Animal motifs	850	40	1	183	21.25	4.5
Human figure motifs	240	4	2	234	60	2
Object motifs	714	33	1	166	21.64	5
Character motifs	150	5	4	85	30	22
Geometric motifs	517	18	1	204	28.72	2.5
Composite motifs	37	11	1	9	3.36	2
Non-motif candidates	281	1	281	281	281	281

2.2. Image Preprocessing

Since the field collected images differed in shooting distance, viewing angle, illumination, and preservation condition (Figure 2), all samples were preprocessed in a unified manner before training. Images with severe blurring, overexposure, repeated shooting, missing subjects, or motif boundaries that were difficult to distinguish were first removed [17]. Samples with clear component sources and relatively identifiable motif subjects were retained. For damaged or occluded samples, images were retained when the main features

could still be recognized. Samples whose categories could not be confirmed were excluded from the training data. The cropped images were uniformly resized to an RGB format of $224 \times 224 \times 3$. To improve robustness under field conditions, ColorJitter and Random Erasing were used during training. ColorJitter was used to simulate outdoor illumination variation, while Random Erasing was used to simulate partial occlusion, stains, and local physical damage. Rotation and perspective transformation were not used, as they may alter the symbolic orientation and compositional order of architectural motifs.



Figure 2. Representative field-image samples showing different spatial carriers and field image conditions.

2.3. Classification System and Annotation Rules

Considering the large number of architectural decorative motif categories in Guanzhong traditional vernacular dwellings, the uneven distribution of samples, and the morphological similarity among some categories, this study adopted a two level classification system to organize motif labels. The first level consists of broad thematic categories, mainly used for category statistics, confusion matrix aggregation, and result interpretation. The second level consists of specific motif names, which serve as the primary label system for model training and testing. Motif category recognition in this study was mainly conducted based on 163 specific categories of architectural decorative motifs. In addition, one non-motif background category was included for rejection training and misrecognition control. Therefore, the complete annotation system contained 164 categories, comprising 163 motif categories and one non-motif category. The non-motif category was not used for typological interpretation, but was used to help the model distinguish decorative motifs from invalid background textures.

The first level included eight categories: plant motifs, animal motifs, human figure motifs, object motifs, character motifs, geometric motifs, composite motifs, and background regions without motifs. For images containing multiple thematic elements, if one element occupied the dominant position, the image was annotated according to that primary element. If multiple elements jointly formed a complete symbolic composition, the image was classified as a composite motif, and the main constituent elements were recorded in the second level label. During annotation, each image sample was assigned a first level category, a second level category, a component location, and an image ID. The second level categories were used for model training and testing, while the first level categories were used for category statistics, confusion matrix aggregation, and result interpretation.

Samples with damaged boundaries or partial occlusion were retained when their main features remained identifiable. Samples with severely missing subjects, indeterminable categories, or potential annotation ambiguity were excluded from the valid motif categories. Through these rules, this study preserved fine grained motif information while establishing a stable label system suitable for recognition tasks involving complex field images.

2.4. Setting of Non-Motif Candidate Region Samples

Misrecognition in complex field images arises not only from the similarity among motif categories, but also from texture interference in non-motif regions. Wall cracks, brick joints, spalled surfaces, water stains, shadows, weathered wood textures, and cluttered backgrounds often present distinct edges or linear textures, making them easily confused with categories such as scroll motifs, water wave motifs, and fret patterns. To reduce false positive predictions, this study introduced non-motif candidate region samples. These samples were not included as architectural decorative motif categories for typological statistics. They were mainly used for candidate region rejection training, enabling the model to learn the features of invalid regions such as wall joints, shadows, stains, spalled surfaces, and natural material textures [18].

3. Recognition Method for Architectural Decorative Motifs

3.1. Overall Framework of the Method

The overall workflow of the proposed method is shown in Figure 3. To reduce the influence of different shooting conditions on model input, scale and color normalization were first applied to the images of Guanzhong traditional vernacular dwellings. SAM was then introduced to generate potential motif candidate regions from complete images, allowing the recognition process to focus first on local areas that may contain decorative motifs. Since wall backgrounds, cracks, shadows, and fragmented textures may still be included in the candidate regions, spatial-prior filtering based on minimum candidate area, minimum side length, aspect ratio, and SAM stability was further used to remove invalid candidate masks. Large candidate regions were not removed by a fixed maximum-area threshold, but were further verified through semantic aggregation. This step removed invalid regions that clearly did not conform to the morphological characteristics of decorative motifs. The filtered candidate regions were cropped into local images and input into the Swin Transformer classification network. Through its hierarchical feature representation capability, the network extracted both local texture features and overall structural information of the motifs, and then predicted category probabilities. Low-confidence candidate results were further removed through the maximum-confidence-based rejection mechanism. The final output included the candidate region location, motif category, and prediction confidence.

3.2. Field Image Preprocessing

To reduce the influence of input differences in field collected images on subsequent candidate region segmentation and classification recognition, scale normalization and color normalization were applied to the images.

For scale normalization, the original image is defined as follows, where H and W denote the image height and width, respectively.

$$I \in \mathbb{R}^{H \times W \times 3} \quad (1)$$

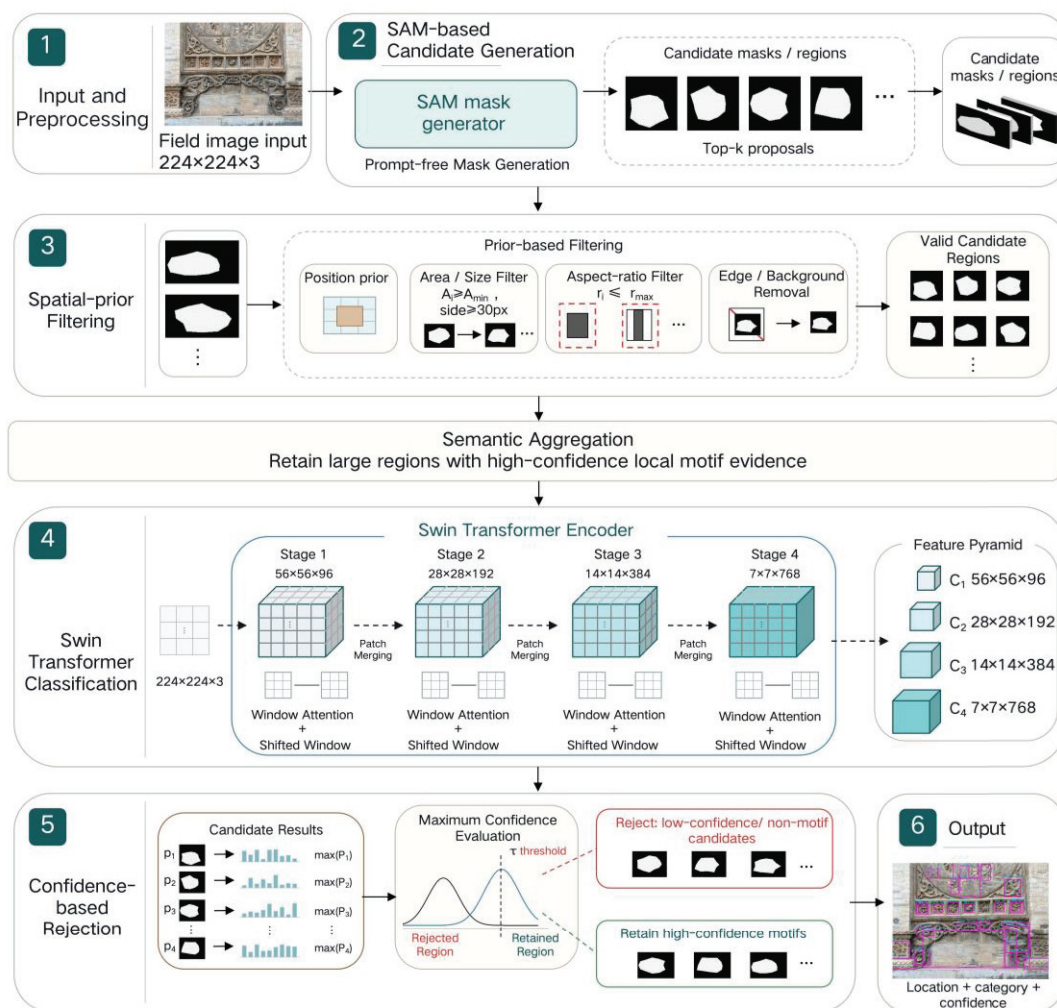


Figure 3. Recognition workflow of architectural decorative motifs from complex field images, including SAM-based candidate generation, spatial-prior filtering, semantic aggregation, Swin Transformer classification, and confidence-based rejection.

To control GPU memory consumption and reduce scale differences caused by different shooting distances, when the long side of an image exceeded the preset threshold L , the image size was adjusted through proportional scaling.

$$s = \frac{L}{\max(H, W)}, I_s = \text{Resize}(I, sH, sW) \quad (2)$$

where s is the scaling factor, and I_s denotes the normalized input image. For color normalization, the RGB channels were standardized as follows.

$$I'_c = \frac{I_c - \mu_c}{\sigma_c} \quad (3)$$

where I'_c denotes the pixel value of the c -th color channel, and μ_c and σ_c denote the mean and standard deviation of the corresponding channel, respectively.

After scale and color normalization, the image was input into the candidate region segmentation module.

3.3. SAM Based Candidate Region Segmentation

To reduce the uncertainty caused by whole image classification, this study introduced SAM as the candidate region generation module [19].

SAM was used to perform full-image candidate region generation on field-collected images. In this study, the SAM model was implemented using the ViT-B checkpoint (sam_vit_b_01ec64.pth). The automatic mask generator parameters were set as follows: points_per_side = 32, pred_iou_thresh = 0.88, stability_score_thresh = 0.92, and min_mask_region_area = 200. Other parameters followed the default settings of the SAM automatic mask generator.

The number of generated masks varied across images depending on component scale, surface texture complexity, carving density, and background fragmentation. These candidate masks were then passed to the subsequent filtering and semantic aggregation modules. This setting allowed the system to generate sufficiently dense candidate regions while reducing unstable masks caused by low-quality segmentation proposals.

$$M = \{m_1, m_2, \dots, m_n\} \quad (4)$$

where m_i denotes the set of candidate masks, and m_i denotes the i -th candidate region. Each candidate mask corresponds to a potential visual region in the image.

3.4. Spatial Prior Filtering

The candidate masks generated by SAM contain a large number of non-motif regions. If these regions are directly input into the classification network, they may increase invalid computation and cause non-motif regions to be misidentified as decorative motifs. To reduce such interference, this study introduced a spatial-prior filtering step after candidate region generation. The filtering process mainly considered candidate area, side length, aspect ratio, and morphological stability.

For each candidate mask m_i , its bounding box is extracted as:

$$B_i = (x_i, y_i, w_i, h_i) \quad (5)$$

where x_i and y_i denote the coordinates of the upper-left corner of the bounding box, and w_i and h_i denote its width and height, respectively. Let A_i denote the area of the i -th candidate mask. In the revised workflow, the area constraint was implemented as an absolute pixel threshold rather than a fixed area-ratio threshold. This setting was adopted because field images varied substantially in shooting distance, component scale, and motif size. A strict area-ratio threshold may remove small but valid decorative motifs. Therefore, candidate masks with an area smaller than 200 pixels were removed. In addition, candidate boxes with a width or height smaller than 30 pixels were excluded because they usually corresponded to tiny fragments, noise, or incomplete texture regions.

The minimum area constraint is expressed as:

$$A_i \geq A_{\min} \quad (6)$$

where A_i denotes the area of the i -th candidate mask, and $A_{\min}$ was set to 200 pixels in the implementation.

The aspect ratio constraint was used to remove elongated regions such as wall cracks, brick joints, and shadow lines. The aspect ratio of the candidate box is defined as:

$$r_i = \frac{\max(w_i, h_i)}{\min(w_i, h_i)} \quad (7)$$

When $r_i > r_{\max}$, the candidate region was regarded as an elongated invalid region and was removed. In this study, $r_{\max}$ was set to 4.0.

The morphological stability constraint was used to remove unstable or fragmented masks. In the revised implementation, this constraint was implemented through SAM `stability_score_thresh`. Candidate masks with a stability score lower than 0.92 were removed. This setting helped exclude fragmented regions such as cracks, spalled edges, and scattered stains.

The final threshold settings used in the revised pipeline are summarized in Table 2.

Table 2. Refined threshold parameters used in the proposed pipeline.

Parameter	Meaning	Final Value in the Revised Pipeline
$A_{\min}$	Minimum candidate area	200 pixels
$I_{\min}$	Minimum width or height of candidate box	30 pixels
$A_{\max}$	Maximum candidate-area threshold	Not used; replaced by semantic aggregation
$r_{\max}$	Maximum aspect ratio	4
$S_{\min}$	Minimum SAM mask stability	0.92, implemented through SAM <code>stability_score_thresh</code>

To further quantify the effect of spatial-prior filtering, this study recorded the number of candidate masks before and after filtering on the complete field-image subset. The SAM automatic mask generator produced an average of 55.33 ± 23.73 raw candidate masks per image. After applying the spatial-prior filtering rules, including minimum candidate area, minimum side length, aspect-ratio constraint, and SAM stability filtering, the average number of retained masks was reduced to 34.13 ± 16.03 per image. These statistics show that the spatial-prior filtering stage removed fragmented or geometrically invalid regions before the subsequent semantic aggregation and classification stages.

As shown in Table 2, the revised pipeline uses absolute minimum-size constraints, aspect-ratio filtering, and SAM stability filtering. A fixed maximum area threshold was not used; instead, large candidate regions were further verified through semantic aggregation based on high-confidence local motif predictions.

Based on these settings, candidate filtering was implemented as a sequential screening process rather than a single rigid mathematical rule. Invalid small, elongated, or unstable masks were removed, while large candidate regions were further verified through semantic aggregation to avoid incorrectly discarding valid decorative regions with large or irregular spatial scales.

3.5. Swin Transformer Based Motif Classification

In this study, the candidate regions were cropped into local images and input into the Swin Transformer classification network for motif category prediction. The recognition of architectural decorative motifs in Guanzhong traditional vernacular dwellings depends on two types of information. The first includes detailed features such as carved lines, edge contours, material textures, and local damage. The second includes structural features such as symmetrical composition, continuous textures, compositional relationships, and overall shape [20]. Swin Transformer extracts detailed information through local window attention and establishes connections between adjacent regions through the shifted window mechanism. It is therefore suitable for processing motif images in which local textures and overall structures coexist.

Let the classification network be $f(\cdot)$, and let the candidate motif image be x . The network output can be expressed as:

$$z = f(x) \quad (8)$$

where z denotes the category score vector output by the classification network. The prediction probability of each category is obtained through the Softmax function:

$$P = \text{Softmax}(f(x)) \quad (9)$$

where $P = \{p_1, p_2, \dots, p_C\}$, C denotes the total number of motif categories, and p_i denotes the probability that the candidate region belongs to the i -th motif category. The model obtains the predicted category of the candidate region according to the category probability distribution, and then passes the result to the subsequent confidence based rejection module.

Through the above processing, the motif recognition task is transformed from direct classification of field collected images into local classification at the candidate region level (Figure 4).

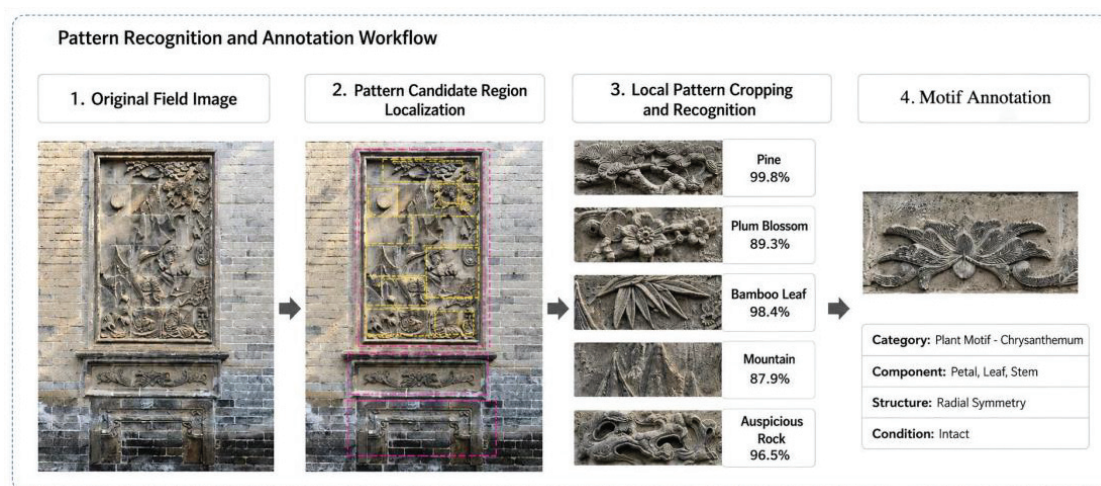


Figure 4. Candidate localization, local cropping, motif recognition, and annotation results on a complete field image.

3.6. Maximum Confidence Based Rejection Mechanism

After candidate region generation and spatial prior filtering, a small number of non-motif regions may still enter the classification network. These regions often have strong edge or texture features and may therefore be assigned to a certain motif category during classification. To reduce this type of misrecognition, this study introduced a maximum confidence based rejection mechanism after classification output to filter low confidence candidate results.

For a candidate region x , the classification network outputs the category probability distribution $P = \{p_1, p_2, \dots, p_C\}$. The maximum prediction probability is defined as:

$$[p_{\max} = \max(P)] \quad (10)$$

The corresponding predicted category is:

$$[y^* = \operatorname{argmax}(P)] \quad (11)$$

When $p_{\max}$ is lower than the preset threshold τ , the model is considered insufficiently reliable in assigning a category to the candidate region, and the system classifies it as background or an invalid region. When $p_{\max}$ is greater than or equal to τ , the candidate region is retained and its predicted category is output. This rule can be expressed as:

$$\hat{y} = \begin{cases} \text{non-motif}, & p_{\max} < \tau \\ \operatorname{argmax}_i P_i, & p_{\max} \geq \tau \end{cases} \quad (12)$$

where τ denotes the maximum confidence rejection threshold, and non-motif denotes a non-motif candidate region or an invalid texture region. The confidence rejection threshold τ was selected on the validation set by comparing macro F1-score, false-positive rate, and candidate recall under different thresholds. In this study, two confidence thresholds were used according to the scale of recognition targets. For macroscopic component anchoring, the confidence threshold was set to $\tau = 0.85$. For microscopic motif recognition, the confidence threshold was set to $\tau = 0.80$. These thresholds were selected according to validation-set performance and were used to balance valid motif retention and non-motif texture rejection. Since category-specific thresholds may introduce instability for rare motif categories, this study adopted unified thresholds and discusses adaptive thresholding as a future improvement. This mechanism reduces false-positive predictions from non-motif textures such as cracks, stains, shadows, and damaged surfaces.

4. Experimental Design and Results

4.1. Experimental Settings and Dwelling-Level Grouped Split

The experiments in this study were conducted using Python 3.11 and PyTorch 2.7.1 with CUDA 11.8 acceleration on an NVIDIA GeForce RTX 4090 GPU (NVIDIA Corporation, Santa Clara, CA, USA). SAM was implemented using the publicly released Segment Anything ViT-B checkpoint, and the YOLOv8 baseline was implemented using YOLOv8n in the Ultralytics YOLOv8 framework. All input images were resized to $224 \times 224 \times 3$ before being fed into the classification network. Cross-Entropy Loss was used for model training, and macro-averaged Precision, Recall, and F1-score were reported to reduce the dominance of frequent categories in performance interpretation. The initial learning rate was set to 1×10^{-4} , and a cosine annealing strategy was adopted to adjust the learning rate. The batch size was set to 32, and the model was trained for 100 epochs. Figure 5 shows the training and validation curves during the first 30 epochs, when the main convergence process occurred. During training, the optimal model parameters were saved according to the macro F1-score on the validation set and the change in validation loss.

To avoid information leakage caused by visually similar images from the same dwelling, component, or motif instance, the experiments were reorganized using a dwelling-level grouped split. All images collected from the same dwelling were assigned to only one subset. For the component-level grouped evaluation, the training set contained 17 dwellings and 2091 images, the validation set contained 6 dwellings and 121 images, and the test set contained 10 held-out dwellings and 351 images. This split refers to the component-level experimental subset used for grouped evaluation, rather than to the entire field-image archive. This grouped split provides a stricter evaluation setting than the previous random split because the test samples come from dwellings that were not seen during training or validation.

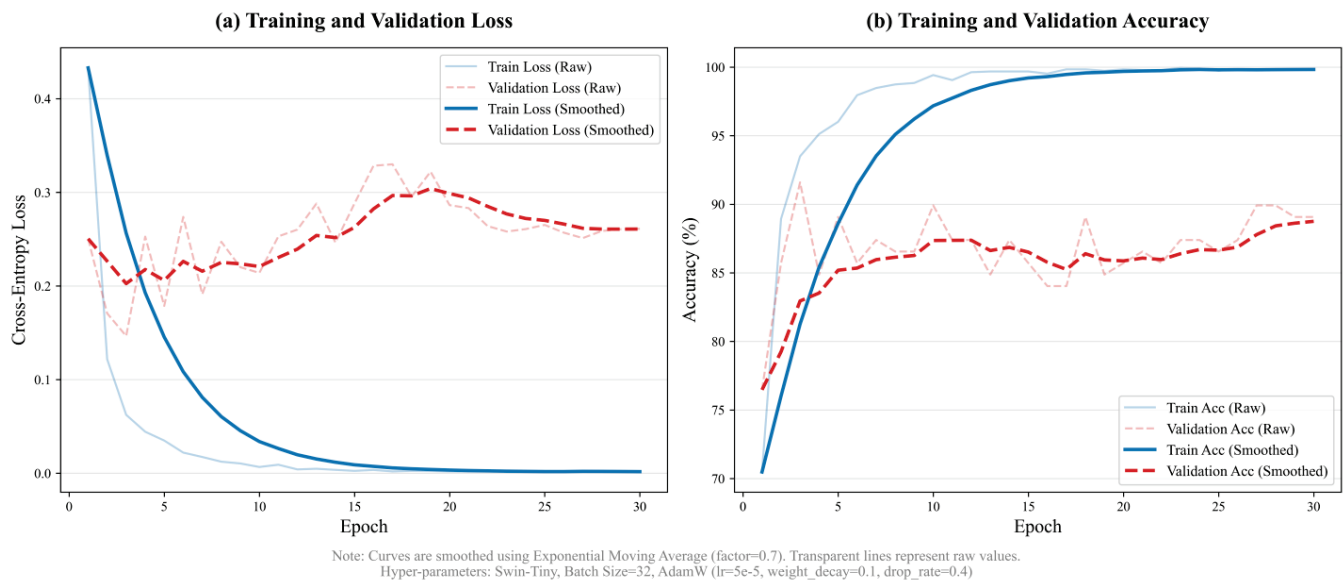


Figure 5. Training and validation curves for the component-level anchoring classifier. (a) Cross-Entropy Loss; (b) accuracy.

The training and validation curves for the component-level anchoring classifier are shown in Figure 5. The classifier converged rapidly during the main training stage, and the validation accuracy remained relatively stable after the early epochs, indicating that the model reached a stable training state under the revised grouped-split setting.

4.2. Evaluation Metrics

The proposed workflow was evaluated at three levels. First, training and validation curves were used to observe the convergence behavior of the classifier. Second, component-level recognition was evaluated on the dwelling-level grouped test set to examine cross-dwelling generalization. Third, the complete workflow was evaluated on full field images using expert verification.

Cross-Entropy Loss was used for multi-class classification. It is defined as:

$$\mathcal{L}_{CE} = -\sum_{k=1}^K y_k \log(\hat{p}_k) \quad (13)$$

where K denotes the number of classes, y_k is the one-hot encoded ground-truth label, and $\hat{p}_k$ denotes the predicted probability of class k .

For the component-level evaluation, Accuracy was calculated as the micro-averaged accuracy over all test samples:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N 1(y_i = \hat{y}_i) \quad (14)$$

where N denotes the total number of test samples, y_i denotes the ground-truth label, $\hat{y}_i$ denotes the predicted label, and $1(\cdot)$ is an indicator function.

For each valid component class k , Precision, Recall, and F1-score were calculated as:

$$P_k = \frac{TP_k}{TP_k + FP_k} \quad (15)$$

$$R_k = \frac{TP_k}{TP_k + FN_k} \quad (16)$$

$$F1_k = \frac{2P_kR_k}{P_k + R_k} \quad (17)$$

where TP_k , FP_k , and FN_k denote the true positives, false positives, and false negatives of class k , respectively.

The macro-averaged Precision, Recall, and F1-score were calculated as:

$$MacroP = \frac{1}{K} \sum_{k=1}^K P_k \quad (18)$$

$$MacroR = \frac{1}{K} \sum_{k=1}^K R_k \quad (19)$$

$$MacroF1 = \frac{1}{K} \sum_{k=1}^K F1_k \quad (20)$$

where K denotes the number of valid component classes. The non-motif category was used during rejection training and false-positive control, but it was excluded from the macro-averaged component recognition metrics. This setting was adopted to evaluate the recognition performance of valid decorative components without inflating the results through the background class.

For complete field-image evaluation, this study used the Expert-Verified Localized Hit Rate (EV-LHR):

$$EV - LHR = \frac{N_{hit}}{N_{total}} \times 100\% \quad (21)$$

where N_{hit} denotes the number of expert-verified localized-hit cases, and N_{total} denotes the total number of complete field images evaluated.

4.3. Candidate Generation and Spatial Filtering Statistics

To quantify the behavior of the SAM candidate-generation stage, this study recorded the number of candidate masks before and after spatial-prior filtering on the complete field-image subset. The SAM automatic mask generator produced an average of 55.33 ± 23.73 raw candidate masks per image. After applying spatial-prior filtering based on minimum candidate area, minimum side length, aspect-ratio constraint, and SAM stability filtering, the average number of retained masks was reduced to 34.13 ± 16.03 per image.

These statistics indicate that the spatial-prior filtering stage removed fragmented or geometrically invalid regions before the subsequent semantic aggregation and classification stages. Since semantic aggregation was used as a subsequent confidence-based verification process rather than as a fixed mask-count filtering threshold, this study reports the quantitative reduction before and after the explicit spatial-prior filtering stage.

The validation performance curves for fine-grained motif recognition are shown in Figure 6. The overall validation accuracy was consistently higher than the macro-averaged metrics, indicating that frequent categories contributed more strongly to the overall accuracy, while rare motif categories reduced macro Precision, macro Recall [21], and macro F1-score. Therefore, macro-averaged metrics were used together with Accuracy to provide a more balanced evaluation under the long-tailed category distribution [22].

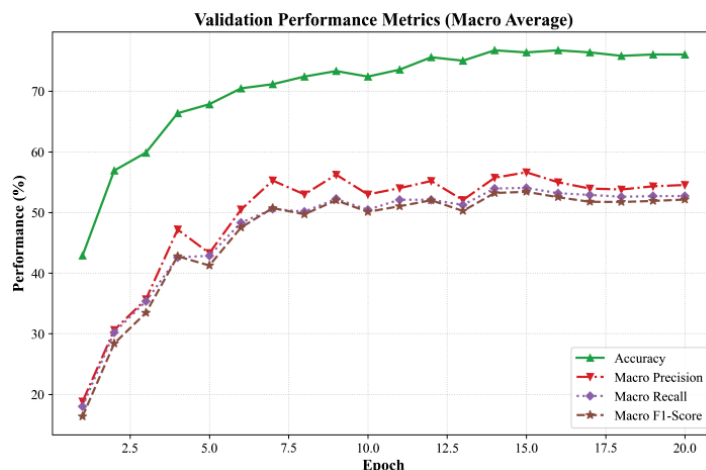


Figure 6. Validation performance curves for the fine-grained motif recognition task under macro-averaged metrics.

4.4. Component-Level Test-Set Evaluation

The component-level recognition performance was recalculated under the dwelling-level grouped test split. The test set contained 351 component-level samples from 10 held-out dwellings and covered six major decorative component categories: drum stone, pillar base, gable decor, screen wall, door pillow stone, and door skirt. Table 3 reports the class-wise performance and overall metrics.

Table 3. Component-level recognition performance under the dwelling-level grouped test split.

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Drum Stone	100	100	100	1
Pillar Base	97.44	82.61	89.41	46
Gable Decor	90	62.5	73.77	72
Screen Wall	94.22	97.02	95.6	168
Door Pillow Stone	67.19	97.73	79.63	44
Door Skirt	83.33	100	90.91	20
Overall accuracy	-	-	88.32	351
Macro average	88.7	89.98	88.22	351
Weighted average	89.78	88.32	88.05	351

Note: Accuracy is micro-averaged over all test samples. Precision, Recall, and F1-score are macro-averaged over the valid component classes. The non-motif category was used for rejection training and false-positive control, but it was excluded from the macro-averaged component recognition metrics.

As shown in Table 3, the proposed method achieved a micro-averaged Accuracy of 88.32%, a macro Precision of 88.70%, a macro Recall of 89.98%, and a macro F1-score of 88.22%. These values replace the previous results obtained under the earlier random split and provide a stricter evaluation of cross-dwelling generalization.

The corresponding component-level confusion matrix is shown in Figure 7. This confusion matrix was computed strictly on the dwelling-level grouped test set, rather than on the validation set or the full evaluated dataset. Most samples were concentrated along the diagonal. The main errors occurred between gable decor, screen wall, and door pillow stone, which may be related to their similar carved surfaces and spatial attachment characteristics in field images.

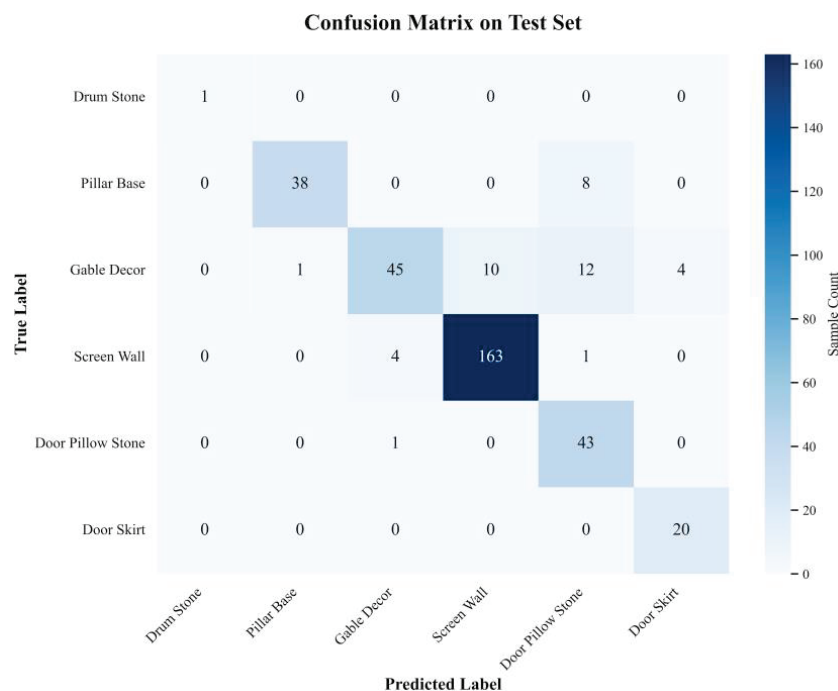


Figure 7. Component-level confusion matrix on the dwelling-level grouped test set.

4.5. Complete Field-Image Evaluation

To evaluate whether the proposed workflow can operate on complete field images rather than only cropped samples, this study further tested 60 complete field images. These images covered typical decorative components, including screen walls, gable decor, door pillow stones, pillar bases, and door skirts, and included common field interferences such as weathering, occlusion, shadows, wall seams, damaged textures, and background clutter. A typical recognition result is shown in Figure 8.

For the complete-field-image evaluation, EV-LHR was calculated through a blind expert-verification protocol. A single senior domain expert in architectural heritage documentation independently evaluated the localized outputs. The expert only viewed the images with predicted boxes and was not informed whether each result was generated by YOLOv8n or by the proposed method. A prediction was counted as a localized hit when the output region covered the main decorative component or motif area and excluded excessive background interference, such as blank walls, wall seams, shadows, and damaged textures. Since the evaluation was conducted by one expert, inter-rater agreement metrics such as Cohen's kappa and disagreement-resolution procedures were not applicable.

For the YOLOv8 baseline, the YOLOv8n variant was used because of its lightweight structure and efficient inference speed. The model was trained for 30 epochs. Its bounding boxes were automatically generated from the first-stage pipeline as high-confidence pseudo-labels through semantic aggregation, without manual bounding-box annotation. During testing, the confidence threshold was set to 0.15 to maximize the recall of the baseline model, and the IoU threshold for non-maximum suppression was set to 0.45. YOLOv8n was evaluated on the same 60 complete field images as the proposed method.

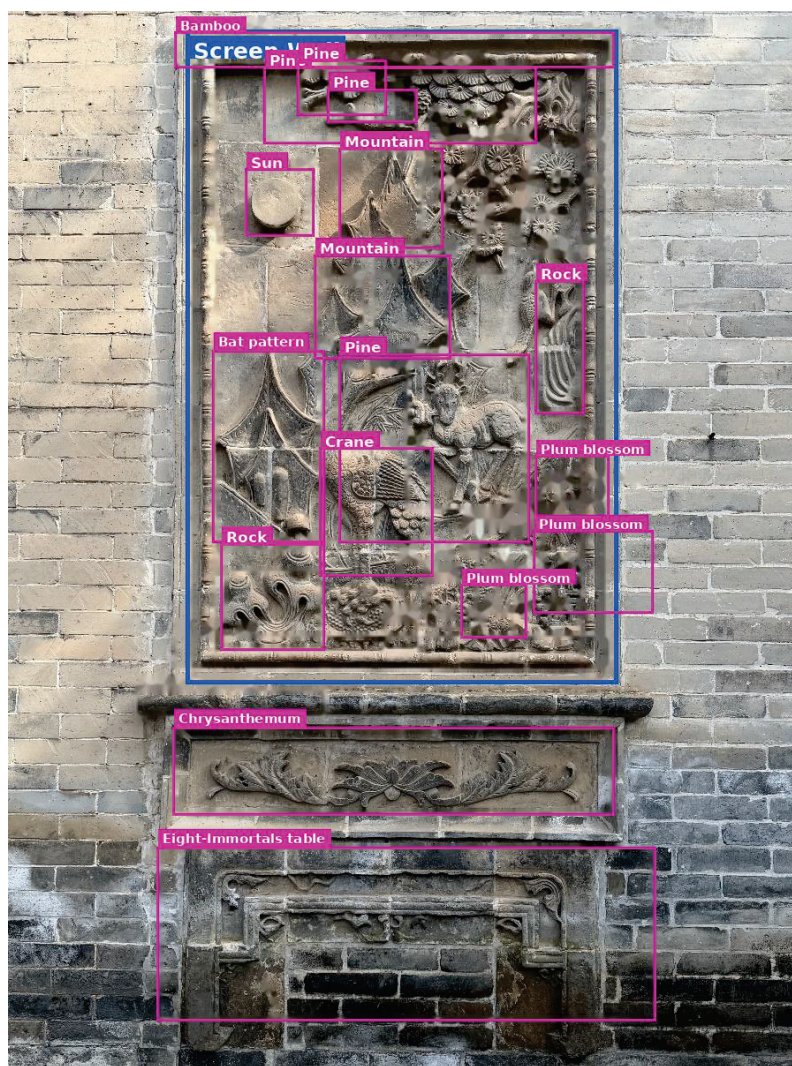


Figure 8. Typical recognition result on a complete screen-wall field image with background-texture interference.

YOLOv8n was evaluated on the same 60 complete field images as the proposed method. The 95% confidence intervals were calculated using the binomial proportion method. The evaluation results are shown in Table 4.

Table 4. Expert-verified localized hit rate on complete field images.

Method	Backbone	Total Test Images	Anchoring Hits	EV-LHR (%)	95% CI
YOLOv8n	CSPDarknet	60	35	58.3	45.9–70.9
Ours	Swin-Tiny	60	41	68.3	56.6–80.0

The proposed workflow achieved an EV-LHR of 68.3% with a 95% confidence interval of 56.6–80.0%, outperforming YOLOv8n by 10.0 percentage points under the same complete-field-image evaluation setting. These results indicate that the proposed method provides more stable macroscopic anchoring in complete field images with wall textures, shadows, weathering, and background interference.

4.6. Module Configuration

To clarify the structure of the proposed workflow, Table 5 summarizes the modules included in each configuration. This table is used to explain the functional composition of the pipeline rather than to report detection-style ablation metrics. Since dense bounding-box or pixel-level annotations were not available for all complete field images, mIoU, AP50, and AP75 were not reported in this study.

Table 5. Module configuration and functional contribution of the proposed workflow.

Variant	SAM	Spatial-Prior Filtering	Semantic Aggregation	Swin Classifier	Confidence Rejection
Swin-T baseline	×	×	×	✓	×
SAM only	✓	×	×	×	×
SAM + filtering	✓	✓	×	×	×
SAM + filtering + semantic aggregation	✓	✓	✓	×	×
Full pipeline	✓	✓	✓	✓	✓

Note: ✓ indicates that the module is included, and × indicates that the module is not included.

As shown in Table 5, the proposed workflow combines SAM-based candidate generation, spatial-prior filtering, semantic aggregation, Swin Transformer classification, and confidence-based rejection. These modules jointly support candidate localization, invalid-region suppression, and final category prediction.

4.7. Limitations

This study still has several limitations. First, dense bounding-box or pixel-level annotations were not available for all complete field images, so detection-style metrics such as mIoU, AP50, and AP75 could not be systematically reported. Second, the complete-field-image evaluation subset contained only 60 images, which limits the statistical generalizability of EV-LHR. Third, EV-LHR is an expert-defined metric, although blind expert verification and confidence intervals were used to reduce subjectivity. Fourth, the full dataset cannot be publicly released because it contains private dwellings, sensitive heritage locations, and unpublished field photographs. Finally, the dataset still has a long-tailed category distribution, and rare motifs remain more difficult to recognize than frequent categories.

5. Conclusions

This study addressed the recognition of decorative motifs in the image based documentation of traditional vernacular architectural heritage. A dataset of architectural decorative motif images from Guanzhong traditional vernacular dwellings was constructed, and a recognition workflow combining SAM-based candidate generation, spatial-prior filtering, semantic aggregation, Swin Transformer classification, and confidence-based rejection was proposed. The workflow reduces interference from non-motif regions such as wall joints, stains, shadows, and weathered textures.

The experimental results show that, under the dwelling-level grouped split, the proposed method achieved a micro-averaged component-level Accuracy of 88.32% and a macro F1-score of 88.22% over valid component classes. The expert-verified evaluation on 60 complete field images further showed an EV-LHR of 68.3%, outperforming the YOLOv8n baseline by 10.0 percentage points. These results indicate that the workflow has practical value for organizing motif regions in complex field images and can provide technical

support for the digital documentation, image retrieval, and conservation management of architectural decorative motifs in traditional vernacular dwellings.

In practical architectural heritage documentation, the proposed workflow can be used as a semi-automatic tool for organizing field images and generating motif-level visual indexes. The system is suitable for batch processing of field photographs, but manual verification remains necessary for rare motifs, heavily damaged components, and ambiguous symbolic compositions. In terms of deployment, the main computational cost is concentrated in SAM-based candidate generation, while the classification and rejection stages are relatively lightweight. When transferred to other regions or other types of architectural heritage motifs, the workflow may require re-annotation of local motif categories, adjustment of spatial-prior thresholds, and additional training samples for region-specific decorative forms.

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

Funding: This research was funded by the National Social Science Fund of China, grant number 24BG160, and the Humanities and Social Sciences Research Project of the Ministry of Education of China, grant number 24XJC760007.

Data Availability Statement: Due to the involvement of private dwellings, sensitive heritage locations, and unpublished field images, the full image dataset is not publicly released. De-identified sample images, annotation protocols, data split files, and key experimental settings are available from the corresponding author upon reasonable academic request. The trained model weights and evaluation code can be provided where permitted by data protection and heritage-site management requirements.

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

References

1. Li, X.K. Three major characteristics of Chinese architectural culture. *Archit. J.* **2001**, *8*, 18–21. (In Chinese)
2. Zhan, Q.C.; Tang, X.J.; Zhao, Y. Innovative application of architectural decorative patterns of ancient dwellings in traditional villages of Guanzhong, Shaanxi in soft furnishing textiles. *China For. Prod. Ind.* **2022**, *59*, 63–68. (In Chinese)
3. Wang, N.; Wu, M.T.; Cao, Z.A.; Xiang, Y. Pattern generation and classification methods for the digital preservation of blue calico. *Packag. Eng.* **2026**, *47*, 201–208. (In Chinese) [CrossRef]
4. Geng, G.H.; He, X.L.; Wang, M.L.; Li, K.; He, X.W. Research progress on key technologies of cultural heritage activation. *J. Image Graph.* **2022**, *27*, 1988–2007. (In Chinese) [CrossRef]
5. Remondino, F.; Rizzi, A. Reality-Based 3D Documentation of Natural and Cultural Heritage Sites: Techniques, Problems, and Examples. *Appl. Geomat.* **2010**, *2*, 85–100. [CrossRef]
6. Remondino, F. Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. *Remote Sens.* **2011**, *3*, 1104–1138. [CrossRef]
7. López, F.J.; Leronés, P.M.; Llamas, J.; Gómez-García-Bermejo, J.; Zalama, E. A Review of Heritage Building Information Modeling (H-BIM). *Multimodal Technol. Interact.* **2018**, *2*, 21. [CrossRef]
8. Tan, G.; Zhu, J.; Chen, Z. Deep learning based identification and interpretability research of traditional village heritage value elements: A case study in Hubei Province. *Herit. Sci.* **2024**, *12*, 200. [CrossRef]
9. Fu, Z.; Yu, G.; Zhang, S.; Zhang, Z. A multi-scale semantic segmentation approach for architectural pattern recognition in traditional Cantonese ancestral halls. *npj Herit. Sci.* **2025**, *13*, 669. [CrossRef]
10. Song, W.T.; Liao, L.Y.; Zhang, H.T.; Li, L.; Yu, T.X.; Zhao, Y.S.; Han, P.Z.; Liu, S.R.; Chen, K.L.; Qu, L.; et al. Applications and prospects of artificial intelligence in the cultural heritage. *J. Image Graph.* **2025**, *30*, 3707–3739. (In Chinese)
11. Fiorucci, M.; Khoroshiltseva, M.; Pontil, M.; Traviglia, A.; Del Bue, A.; James, S. Machine Learning for Cultural Heritage: A Survey. *Pattern Recognit. Lett.* **2020**, *133*, 102–108. [CrossRef]

12. Llamas, J.; Leronés, P.M.; Medina, R.; Zalama, E.; Gómez-García-Bermejo, J. Classification of Architectural Heritage Images Using Deep Learning Techniques. *Appl. Sci.* **2017**, *7*, 992. [CrossRef]
13. Croce, V.; Caroti, G.; Piemonte, A.; De Luca, L.; Véron, P. H-BIM and Artificial Intelligence: Classification of Architectural Heritage for Semi-Automatic Scan-to-BIM Reconstruction. *Sensors* **2023**, *23*, 2497. [CrossRef] [PubMed]
14. 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*; IEEE: New York, NY, USA, 2016; pp. 770–778. [CrossRef]
15. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021*.
16. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Conference, 11–17 October 2021*; IEEE: New York, NY, USA, 2021; pp. 10012–10022. [CrossRef]
17. Shorten, C.; Khoshgohar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
18. Gao, F.; Yang, L.; Li, H. A survey of open set recognition. *J. Nanjing Univ.* **2022**, *58*, 115–134. (In Chinese) [CrossRef]
19. Zhang, T.S.; Zhu, M.F.; Ren, Y.W.; Chen, W.; Zhou, K.; Xu, K.; Bao, H.J. BPA-SAM: Box prompt augmented SAM for traditional Chinese realistic painting. *J. Graph.* **2025**, *46*, 322–331. (In Chinese) [CrossRef]
20. Zong, L.C.; Wang, N.N. Research on a computer-aided method for identifying feature lines of bronze decorative patterns. *J. Graph.* **2020**, *41*, 1018–1023. (In Chinese) [CrossRef]
21. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
22. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 9260–9269. [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.

Article

A Testability Strategy Optimization Method Under Multi-Valued Dependency Condition Based on Deep Reinforcement Learning

Chao Zhang ^{1,2,*}, Yufei Zhang ¹, Feng Wang ³, Xiaoxu Su ⁴, Zhijie Dong ⁴ and Linlin Zuo ⁴

¹ Department of Integrated Technology and Control Engineering, School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China; nendneke@mail.nwpu.edu.cn

² National Key Laboratory of Aircraft Configuration Design, Xi'an 710072, China

³ Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China; fenfen@mail.nwpu.edu.cn

⁴ The 6th Research Institute of China Electronics Corporation, Beijing 102209, China; suxiaoxu@ncse.com.cn (X.S.); dongzhj@ncse.com.cn (Z.D.); hhhlpbo@sohu.com (L.Z.)

* Correspondence: caec_zc@nwpu.edu.cn

Abstract

The multi-valued dependency matrix (MVD matrix) is an important testability modeling approach, which can deliver more comprehensive testability information than the traditional dependency matrix (D-matrix). However, existing testability strategy optimization algorithms perform poorly in handling the MVD matrix, and the high-dimensional MVD matrix further aggravates these limitations as system complexity increases. To address these problems, a novel testability strategy optimization method under multi-valued dependency conditions based on deep reinforcement learning (DRL) is proposed. Firstly, the sets of elements and two reward functions to minimize test sequence length and test cost are established from the MVD matrix. Subsequently, the algorithm for selecting test points based on Deep Q-Network (DQN) is proposed. The DQN parameters are updated to fit the Q-value of test points. Thirdly, Double DQN (DDQN) and the prioritized experience replay (PER) mechanism are introduced to address the overestimation problem and sample redundancy problem, respectively, in high-dimensional matrix environments. The experimental results show that the testability strategy generated by this method can isolate all faults with fewer steps or at a lower cost. In a high-dimensional matrix environment, it can reduce test costs compared with the other heuristic algorithms while maintaining a good level of stability.

Keywords: multi-valued dependency matrix; testability strategy optimization; deep reinforcement learning; deep Q-network

1. Introduction

Testability is an inherent characteristic of a product, which refers to the ability to conveniently, quickly, and accurately determine its state and isolate internal faults under specified conditions and within a given time frame. The main purpose of testability is to shorten the time or cost for fault isolation, thereby ensuring the reliability and maintainability of the product [1–4].

In testability modeling, the dependency matrix (D-matrix) is the most widely used and the most intuitively described method [5–8]. The traditional single-value D-matrix uses binary logical values, with two Boolean logics respectively indicating whether the test point

can detect the fault [9–11]. For complex systems, due to the extremely large number of faults and test points, the testing efficiency of the traditional D-matrix is often very low [12]. A dependency matrix with multiple test values is referred to as a multi-valued dependency matrix (MVD matrix) [13], and it reflects the relationship between faults and tests in a multi-valued form. Currently, testability modeling based on the MVD matrix method has been widely applied in various fields. Huang et al. [14] classified the relationship between faults and test points into three categories, thereby improving the isolation efficiency of the MVD matrix. Su et al. [15] conducted MVD matrix-based fault testability analysis for engine systems. Zhu et al. [16] proposed a new method for constructing the four-valued D-matrix to conduct testability modeling for the fuel metering system of the aero-engine. The use of an MVD matrix enables the acquisition of richer test result information and improves the test efficiency of the model.

The primary role of the D-matrix is to assist testers in formulating an optimal testability strategy, thereby enabling the corresponding test point to isolate all faults with fewer steps or lower costs [17,18]. Currently, many scholars have conducted relevant studies on this issue [19–21]. For instance, Liang et al. [22] proposed an optimal testability strategy method based on the partially observable Markov decision process (POMDP) to address the problem of unreliable test strategy. Cui et al. [23] introduced hash functions to convert the dependency matrix into key-value pairs, and utilized the binary search algorithm to locate the required fault modes. Zhang et al. [24] combined the advantages of the classic algorithms' information gain and fault detection weight index to improve the optimization efficiency. Wang et al. [25] solved the complex system fault test strategy problem using swarm intelligence algorithms, resulting in an 8–10 times improvement in testability performance.

However, these methods are only applicable to the traditional binary D-matrix and have not been extended to the MVD matrix level. When dealing with multi-valued testing, many of these methods either become inapplicable or yield poor results. Currently, there are relatively few studies on optimizing testability strategy based on the MVD matrix, and the existing research mainly focuses on the application of heuristic algorithms [15,16]. Dandan et al. [26] proposed a multi-value testability strategy optimization method based on information entropy, but this greedy algorithm is prone to getting stuck in a local optimal solution. Tian et al. [27] introduced the swarm optimization algorithm into the solution of the multi-valued system test strategy, but the algorithm is limited by its computational complexity. As modern equipment becomes larger and more complex, the dimensionality of the MVD matrix increases sharply. Consequently, the inherent limitations of existing optimization methods become more pronounced. This leads to an exponential growth in computational costs when existing algorithms are applied to complex systems, which not only makes it difficult to obtain the optimal solution, but also results in excessively long operation time.

Deep reinforcement learning (DRL) based on the Markov decision model is an integrated technology combining deep learning and reinforcement learning, and its core idea is to enable an agent to independently explore the optimal strategy through “trial-and-error learning” in a complex environment [28,29]. Moreover, it can also utilize the deep neural network to perform value fitting, thereby addressing the combinatorial explosion problem in the state space [30]. Currently, DRL has been widely applied in fields such as target recognition [31], path planning [32], and intelligent decision-making [33].

The core of the testability strategy solution based on the MVD matrix lies in the decision-making and selection process of test points, which can be regarded as a Markov decision process. DRL can learn the optimal action plan more quickly, which essentially means exploring the optimal testability strategy. When facing the high-dimensional MVD

matrix, DRL can significantly enhance the efficiency and basic accuracy of the testability strategy optimization process compared with traditional algorithms, and can also provide a more reliable test strategy.

Therefore, this paper proposes a novel testability strategy optimization method based on the idea of deep reinforcement learning. Taking the sampling circuit and the multi-functional integrated radio frequency system (MIRFS) as the research objects, respectively, the feasibility of this method is verified. Firstly, the Markov decision model is established based on different optimization goals of the testability strategy to determine the reward direction for the intelligent agent. Secondly, a Deep Q-Network (DQN) is proposed to select the optimal test point under different fault isolation states. Thirdly, Double DQN (DDQN) and prioritized experience replay (PER) mechanisms are introduced, respectively. The model trained can output the optimal test strategy and diagnostic tree. The main contributions of this paper are as follows:

- (1) The Markov decision model for solving the testability strategy is established. Firstly, the intelligent agent, state set, action set and state transition probability are defined from the MVD matrix. Then, the reward functions for testability strategy optimization are established, and the test sequence length or test cost is linked to the reward functions. Based on the idea of accumulating the maximum reward in a Markov decision process, the agent will strive to minimize the length of the test sequence or the test cost while isolating all faults.
- (2) A novel test point optimization algorithm based on the DQN is proposed. A deep neural network is used to approximate the Q-value functions, and the selection of the test point will be based on the neural network. Experience samples are obtained through the MVD matrix, and the parameters of the Q-network are moderately updated through sample learning based on the Markov decision model. The update of parameters enables the fitting of the Q-value of test points to converge towards the shortest test sequence or the lowest test cost.
- (3) DDQN and PER mechanisms are introduced to mitigate overestimation and redundant experience samples problems, respectively. As the MVD matrix structure continues to expand, the overestimation problem of the DQN will be severe. Meanwhile, the experience samples will also become complex and redundant, which leads to a decrease in the training efficiency of the model. Therefore, the DDQN and the PER mechanism are respectively introduced to construct the Prioritized Experience Double Deep Q-Network (PE-DDQN) algorithm model. It can alleviate the overestimation problem and make the algorithm learn the optimal action plan more effectively.

The experimental results show that the testability strategy optimization method proposed in this paper can effectively isolate all faults with fewer steps or at a lower cost. In a contrast experiment conducted in a high-dimensional matrix environment, it can reduce test costs compared with the other heuristic algorithms while maintaining a good level of stability. For high-dimensional MVD matrices of complex systems, this method has strong engineering application prospects and promotion value.

The core contribution of this study lies in the development of a novel framework based on the existing DRL algorithms, which is structurally tailored for testability strategy optimization under multi-valued dependency conditions. By strategically integrating mature DRL techniques into the generation of testability strategies, this work significantly enhances the applicability of modern AI algorithms in solving practical engineering testing problems.

2. Problem Description and Model Construction

2.1. Problem Description

The test strategy optimization problem is a process aimed at designing a series of tests with a specific strategy based on a predefined quintuple. The core objective is to effectively identify and isolate the potential fault state set $F = \{f_1, f_2, \dots, f_m\}$ using a strategically designed sequence of $T = \{t_1, t_2, \dots, t_k\}$.

2.1.1. MVD-Matrix

The MVD matrix is an extended form of the D-matrix for the analog signal test, which represents the matrix element values as the number of different fault fuzzy groups according to the effective value of test signals [34,35]. Its row vectors represent the test values of a certain fault at multiple test points, and a fault is determined when the actual test values completely match the corresponding row vector. The MVD matrix has an element value range of 0 to k-1, which is shown as

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{bmatrix} \quad (1)$$

When constructing the MVD matrix, the following two sets need to be determined first:

(1) Fault Set

$F = \{f_1, f_2, \dots, f_m\}$ represents the set of possible failure modes that may occur in the entire system, where m denotes the total number of failure modes. Each failure f_i represents an independent failure that may cause functional failure in the system.

(2) Test Set

$T = \{t_1, t_2, \dots, t_k\}$ represents the set of available test items in the system, where k denotes the number of test items. Each test t_j is a tool used to detect or verify a certain fault state.

On this basis, the MVD matrix needs to be expanded with two additional parts:

(1) Prior Probability Set:

$P = \{p_1, p_2, \dots, p_m\}$ represents the prior probability associated with each fault f_i in the fault state set, and the sum of these probabilities is 1.

(2) Cost Set:

$C = \{c_1, c_2, \dots, c_n\}$ represents the cost incurred by using the test point t_i , which includes the manpower, material resources, financial resources, and time consumed.

2.1.2. Optimization of Testability Strategy

Isolating all faults rarely requires utilizing every test point. Due to inherent redundancy, selecting a targeted subset is usually sufficient. Meanwhile, the sequence order also has an impact on the isolation of faults. Under different sequence orders, the sequence of fault isolation and the speed of isolation will vary, which will affect the speed of troubleshooting when equipment fails. Furthermore, the testing plan can be dynamically adjusted based on the outcome of each test, which is the primary purpose of a testing strategy. The testability strategy can also be depicted in the form of a diagnostic tree.

For the optimization of the testability strategy, there are two specific directions:

(1) The test sequence that isolates all faults contains the fewest test points. The elements of the set T_f of the fault isolation set F are the fewest, as shown in

$$L = |T_f|. \quad (2)$$

This optimization direction focuses on the minimum identification steps for fault isolation. Based on the assumption of the homogeneity of test operation costs, it achieves rapid fault location by constructing the smallest independent test set. Essentially, it is the solution to the smallest complete test subset within the framework of combinatorial optimization, and is applicable to scenarios where test resources are homogeneous and the constraints are mainly time complexity.

(2) When all faults are isolated, the weighted test cost is minimized. The calculation formula for the test cost is shown as

$$J = \sum_{i=1}^m \left(\sum_{j=1}^{|T_{f_i}|} c_{T_{f_i}[j]} \right) p(f_i), \quad (3)$$

where $p(f_i)$ is the prior probability of fault i ; T_{f_i} is the test sequence used for isolating fault f_i ; $|T_{f_i}|$ is the length of the test sequence; $c_{T_{f_i}[j]}$ is the cost of the j -th test in the test sequence list for isolating fault f_i . In order to make the expected test cost universal, this paper sets the test cost $c_{T_{f_i}[j]}$ as a dimensionless physical quantity.

This direction involves expected cost optimization under probability weighting. The cost is weighted by the prior probability of failure, and the heterogeneity costs of the testing operations are considered. Through Bayesian inference and dynamic programming, the statistical average cost of fault isolation is minimized, corresponding to scenarios such as the full life cycle maintenance of complex equipment and the allocation of high-cost testing resources.

Our goal is to minimize the test sequence length or the test cost while isolating all faults.

2.2. Model Construction

With the increase in fault types and test points, the dimensions of the MVD matrix model become extremely large, and traditional optimization methods perform poorly. The DRL optimization method is based on Markov decision processes (MDPs) [36], and it can find more efficient and accurate solutions, which provides a new approach for the test optimization of multi-valued systems. Fundamentally, the exact underlying fault in the system is hidden, which traditionally formulates the test strategy optimization problem as a POMDP. However, to efficiently apply DRL, we resolve this by utilizing the concept of information state. By tracking the current set of suspected faults based on past test outcomes, the next ambiguity state strictly depends only on the current ambiguity state and the selected test. Therefore, we reformulate the problem into a fully observable MDP over the information space. This provides a rigorous and consistent theoretical basis for our DRL framework. In this section, we establish an MDP model based on the MVD matrix and combine this model with the optimization strategy to form a complete framework for solving the testability strategy optimization problem.

2.2.1. Set of Elements

(1) Agent:

The agent leverages accumulated experience to maximize rewards and learns from past suboptimal choices. In this framework, the agent's actions correspond to the selection of test points, aiming to choose the optimal test point in any given state.

(2) State set

Assume the current system's MVD matrix contains m fault modes and n test points, with multi-valued elements ranging from 0 to $k - 1$. We strictly define the agent's state as a three-dimensional tensor. The principal difficulty in applying reinforcement learning

to high-dimensional MVD matrices lies in the definition of states: as the matrix grows, the state space becomes high-dimensional and only partially predictive of the eventual isolation reward, a challenge identified as central to deploying reinforcement learning beyond tractable laboratory conditions, where states are high-dimensional, continuous, and partially observable [37]. The three-channel state encoding adopted here is one attempt to expose the reward-relevant subset of state dimensions to the network.

The three channels extract data from different physical and logical dimensions of the environment:

Channel 1: static MVD matrix

This serves as the foundational base map of the environment, providing the network with the global static test response patterns.

$$C_{i,j}^{(1)} = \frac{d_{i,j}}{k-1}, \quad (4)$$

where $d_{i,j}$ represents the original matrix element, normalized by dividing by the maximum value $k-1$ to ensure a stable input distribution for the network.

Channel 2: dynamic fault row-mask

This channel reflects the real-time “survival status” and “fuzziness” of each fault, guiding the convolution kernels to focus on rows that have not yet been isolated.

$$C_{i,j}^{(2)} = \begin{cases} 0, & \text{if fault } f_i \text{ is completely isolated} \\ \frac{|g_x|}{m}, & \text{if fault } f_i \text{ remains in fuzzy group } g_x \end{cases}, \quad (5)$$

where rows of isolated faults are zeroed out entirely. Faults still in a fuzzy group are assigned the relative size of that group. By encoding the dynamic survival status of the faults into this channel, the agent is provided with a fully observable belief state, perfectly aligning with the aforementioned MDP formulation.

Channel 3: action column-mask

This records the consumption of the test sequence, directly cutting off the feature transmission of executed actions within the 2D space.

$$C_{i,j}^{(3)} = \begin{cases} -1, & \text{if test point } t_j \text{ has been executed} \\ 1, & \text{if test point } t_j \text{ has not been executed} \end{cases}, \quad (6)$$

where executed test columns are entirely set to -1 . Paired with the subsequent activation function, this creates an excellent dead zone masking effect, preventing the network from evaluating already-taken actions.

(3) Action set

Action set represents the actions that the agent can take in each state. Actions are usually represented by the symbol A . In this model, the elements of the action set are defined as the execution of individual tests, and it can be expressed as

$$A = T = \{t_1, t_2, \dots, t_n\}, \quad (7)$$

where n represents the total number of test points in the MVD matrix.

(4) State transition probability

The state transition probability describes the probability of moving to another state after taking a specific action in a certain state. It is usually denoted by $P(s_{j+1}|s_j, a_j)$, which represents the probability of transitioning from state s_j to state s_{j+1} given action a_j .

Under the optimization goal of sequence length, we only need to consider the minimality of the test sequence. By default, each test will be conducted on all the fuzzy groups. Therefore, the state transition is directly defined by the changes in the fuzzy groups before and after the test, thereby forming a deterministic transition probability. In contrast, under the cost optimization objective, different branches can adopt different testing schemes. The environment simulates a single test response based on the faults' prior probability distribution, directing the system to a specific fault subset. The model will form a unique subsequent state based on this subset of fuzzy groups. This MVD matrix-driven physical transition prevents the agent from falling into superposition states and ensures it learns the true environmental dynamics, indicating a probabilistic rather than fixed state transition mechanism.

The relationship diagram of the Markov model obtained from the MVD matrix is shown in Figure 1.

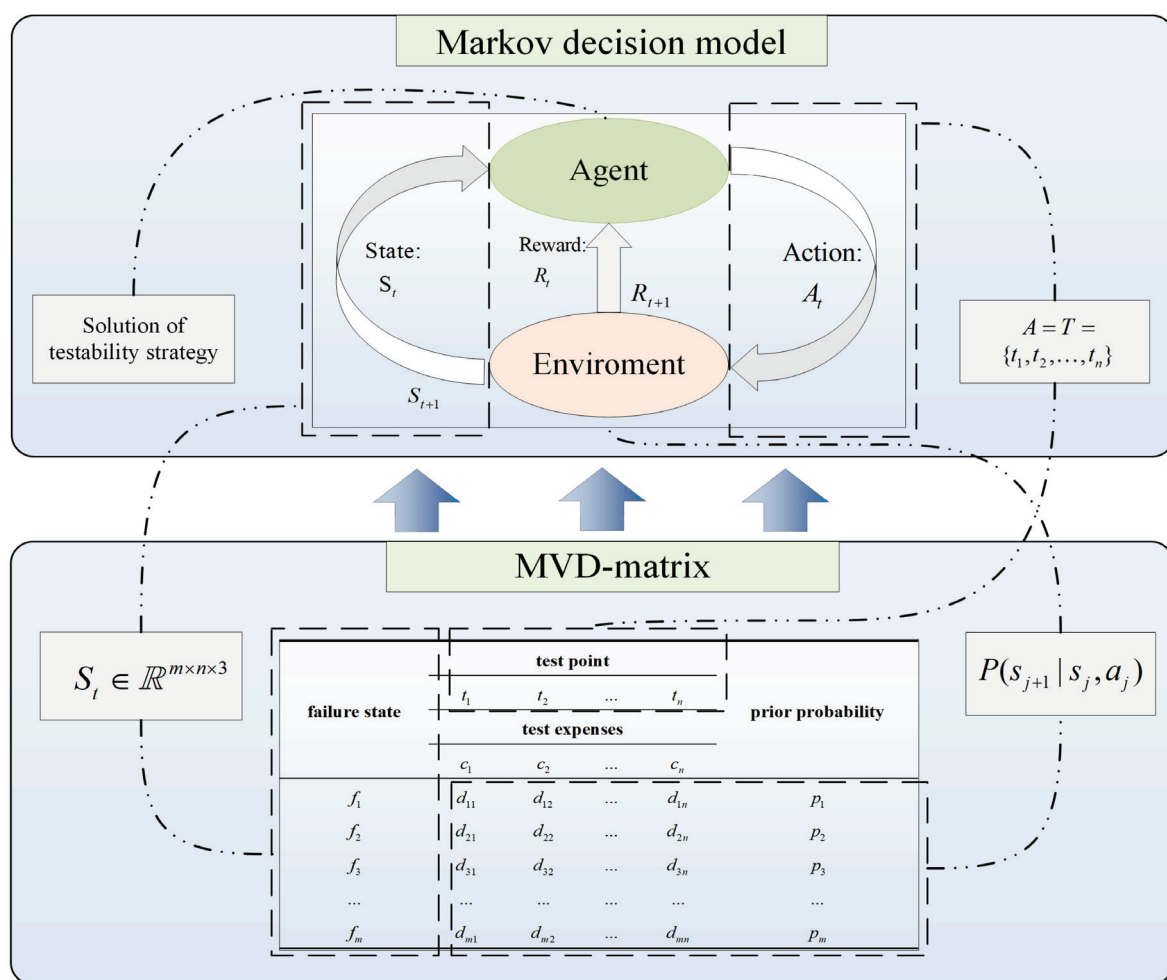


Figure 1. Model construction process.

2.2.2. Reward Function

For the Markov decision process, the goal of the agent is to accumulate the maximum reward. The reward function $R(s_j, a_j, s_{j+1})$ represents the reward obtained when taking action a_j and transitioning from state s_j to state s_{j+1} . By executing an action, the agent will transition from one state to another. After executing an action in a certain state, the agent will receive a reward. Through continuous trial and error and iterative updates, eventually, the scores of each action in different states are obtained. Based on these score values, testability strategy decisions will be generated.

(1) The shortest test sequence

For the shortest test sequence, the classic constant step penalty method is adopted. By using dense negative feedback, the intelligent agent is forced to search for the shortest path. The basic idea is that when the agent isolates all the fault fuzzy groups, a positive reward is given; otherwise, a small negative reward will be given. This indicates that when the agent moves from state s_j to state s_{j+1} by taking action a_j , if the total number of completely isolated faults in state s_{j+1} reaches the total number of faults, a positive reward is given, and a small negative reward is given in other states. At the same time, when the agent's state reaches the above condition, it means that all fault isolation has been completed, and the exploration ends at this point. This reward mechanism ensures that the agent prioritizes ensuring that only the shortest test sequence can isolate all fault groups, thereby achieving optimality in the direction of the shortest test sequence. Its reward function is shown as

$$R(s_j, a_j, s_{j+1}) = \begin{cases} r_{pos}, & G_{total} = m \\ -\varepsilon, & other \end{cases}, \quad (8)$$

where r_p represents a large positive terminal reward for complete fault isolation; ε is a strictly positive scalar representing a small fixed step penalty; and m represents the total number of faults. This mechanism ensures that the agent pays a consistent penalty $-\varepsilon$ for every redundant or inefficient test, thereby driving the Q-learning process to prioritize the absolute shortest path to reach r_p .

(2) The minimum test cost

Regarding the minimum testing cost, since directly using negative reward costs would lead to inconsistent punishment scales and exploration failure, an innovative cost savings margin method is proposed, which can utilize a purely forward incentive without negative feedback to guide the network to converge. The rewards rules are as follows:

(1) For any given state transition, if the action a_j fails to isolate any faults in the currently explored fuzzy group, no reward is given.

(2) When the current fuzzy group is completely isolated down to a single specific fault f_i , the current episodic exploration terminates. According to Equation (3), a terminal reward $R_{terminal}$ is given based on the cost savings margin. To calculate this, let T_{seq} represent the sequence of tests executed by the agent in the current episode. The terminal reward is formulated as

$$R_{terminal} = \eta \left(C_{total} - \sum_{t_j \in T_{seq}} c_j \right), \quad (9)$$

where c_j is the cost associated with executing test t_j . As established in Section 2.1.2, both c_j and C_{total} are treated as dimensionless physical quantities to ensure universal applicability across different system parameters. The parameter η is a positive reward scaling coefficient.

Consequently, the step-by-step reward function R_t is shown as

$$R_t = \begin{cases} R_{terminal}, & \text{if isolated to single } f_i \\ 0, & \text{otherwise} \end{cases}. \quad (10)$$

While standard pathfinding algorithms often rely on negative step penalties, this cost optimization model uniquely achieves convergence using a purely forward, non-negative incentive. Because the fault isolation process is framed as a strictly finite-horizon episodic task, an inverse-cost terminal payout is mathematically sufficient to guarantee policy convergence. By ensuring that taking fewer, cheaper tests results in a strictly larger $R_{terminal}$,

the optimal policy naturally aligns with the minimum-cost sequence. Furthermore, the coefficient η acts as a crucial scaling parameter. Empirical tuning of subsequent experiments indicates that establishing η to map the terminal reward into a moderate numerical range (e.g., 10^1 to 10^2 as reflected in the stable Q-value convergence curves in Section 4) yields the highest training stability. This constraint prevents extreme magnitude variance in the target network, effectively mitigating gradient explosion or vanishing, and ensuring the smooth convergence of the PE-DDQN model while maintaining sufficient numerical resolution to distinguish subtle cost differences.

During the output process of the diagnostic tree, when optimizing the testing cost, the cost generated by isolating a fuzzy group through multiple tests is different. In this case, it is stipulated that each fuzzy group can use different testing methods. Therefore, for each unisolated fuzzy group, a recursive state traversal mechanism is adopted. Each fuzzy group constructs an independent set of state spaces and independently selects the optimal action to ensure that the weighted cost can be minimized.

3. The Proposed Testability Strategy Optimization Algorithm

After constructing the element set and reward functions in the Markov decision model based on the MVD matrix, it is necessary to use an effective DRL algorithm to solve the Markov decision problem. This section proposes a solution algorithm based on the DQN. By continuously obtaining samples from the MVD matrix to train the DQN and updating its parameters, the network accurately estimates the Q-value of test points in each state, providing a basis for selecting test points. At the same time, the PER and DDQN are introduced to solve the overestimation problem and the problem of excessive sample size that occur during the model training process due to the high-dimensional MVD matrix. The DQN trained by the two proposed reward mechanisms can respectively generate test strategies and diagnostic trees for the two optimization goals. The flowchart of solving the testability strategy for the MVD matrix is shown in Figure 2.

3.1. The Test Point Optimization Strategy Based on DQN

During the test point selection process, the agent's goal is to maximize the cumulative reward. Therefore, it needs to add the maximum reward that can be obtained in future states to the reward for achieving the current state. The core of the algorithm utilizes the Bellman equation as the value iteration update [38,39], which is shown as

$$Q^{\text{new}}(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)], \quad (11)$$

where $Q^{\text{new}}(s, a)$ is the updated Q-value of the agent's state s for action a ; $Q(s, a)$ is the Q-value of the intelligent agent's state s and action a before the update; $Q(s', a')$ is the next state s' of the updated agent, along with the Q-value for each possible action; r is the reward obtained by the current action a interacting to transform into the next state s' ; α is the learning rate; γ is the discount factor.

The DQN algorithm uses certain deep neural networks to approximate the Q-value functions based on Q-learning [40,41]. Figure 3 presents a reinforcement learning training framework for test point selection based on DQN. The environment is represented by the MVD matrix, which includes fault states, test points, detection capabilities, and costs. DQN employs a greedy strategy to obtain experiences from the MVD matrix and conducts model parameter training. After the DQN model is trained, the initial state is input into the trained DQN and continuously selects the test point with the highest score to generate a testability strategy. Based on this, the steps for building DQN to solve the testability strategy are as follows:

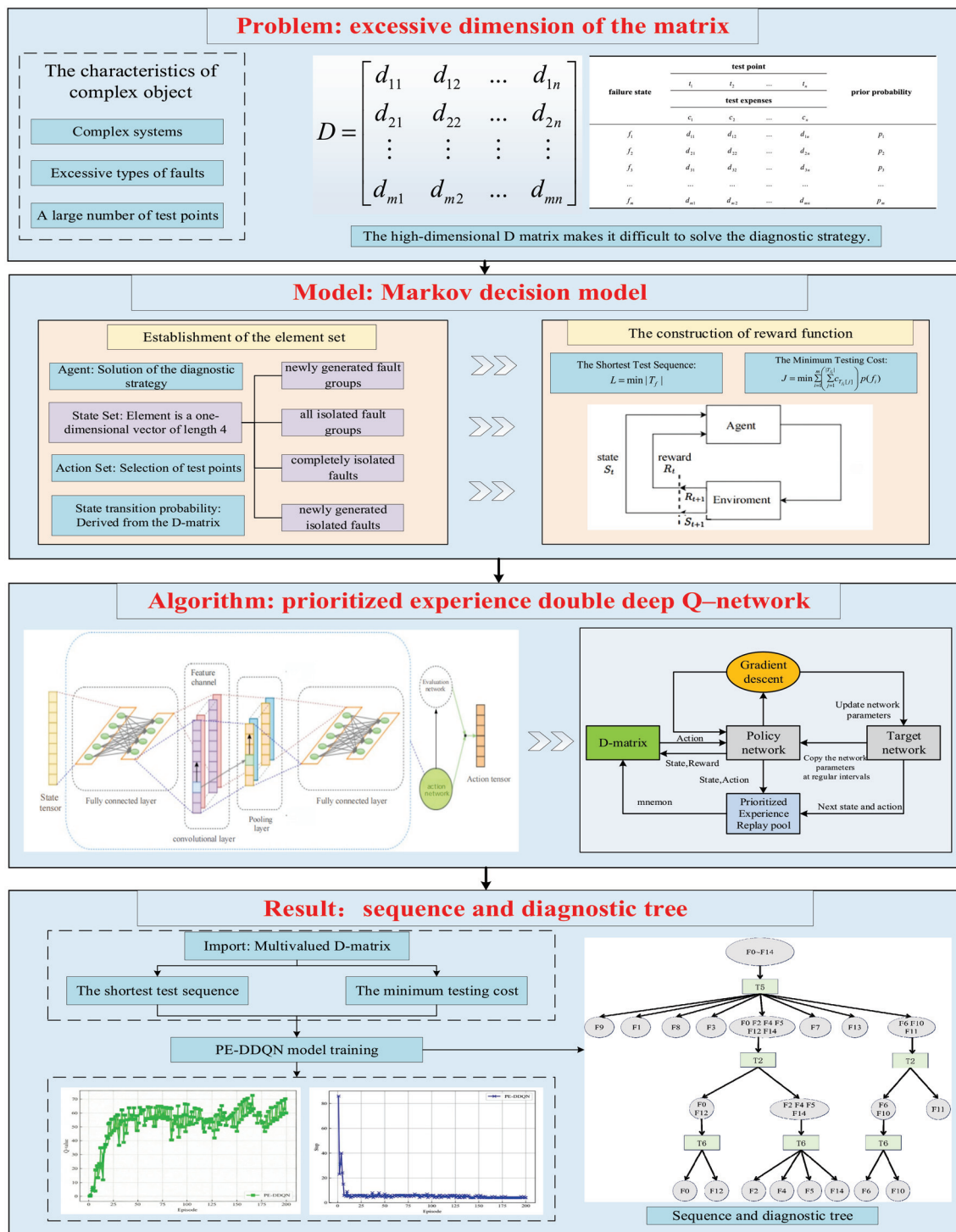


Figure 2. Testability strategy solution flowchart.

Step 1. Initialization of network and parameters

To stabilize the training process, the standard DQN algorithm utilizes two neural networks with identical structures but distinct parameter sets. Firstly, the DQN is initialized, including the policy network Q_1 and the target network Q_2 . The network parameters are defined: let θ denote the parameters of the policy Q-network, which is continually updated at each step; let θ^- denote the parameters of the target Q-network, which is used to calculate the target value and is only updated periodically. The networks consist of convolutional layers, pooling layers, and fully connected layers, as shown in Figure 3. The networks take the three-channel state space vector as the input, and further capture

the coupling relationship between the number of fault groups and the isolation efficiency through convolution-pooling operations. The fully connected layer is the last layer of the convolutional neural network, which is used to fuse and classify the features extracted by the convolutional neural network. In DQN, each element of the output vector corresponds to the Q-value estimate of an action. The Q-value estimates obtained by flattening the output of the previous layer of the fully connected layer are the basis for the agent to make action decisions.

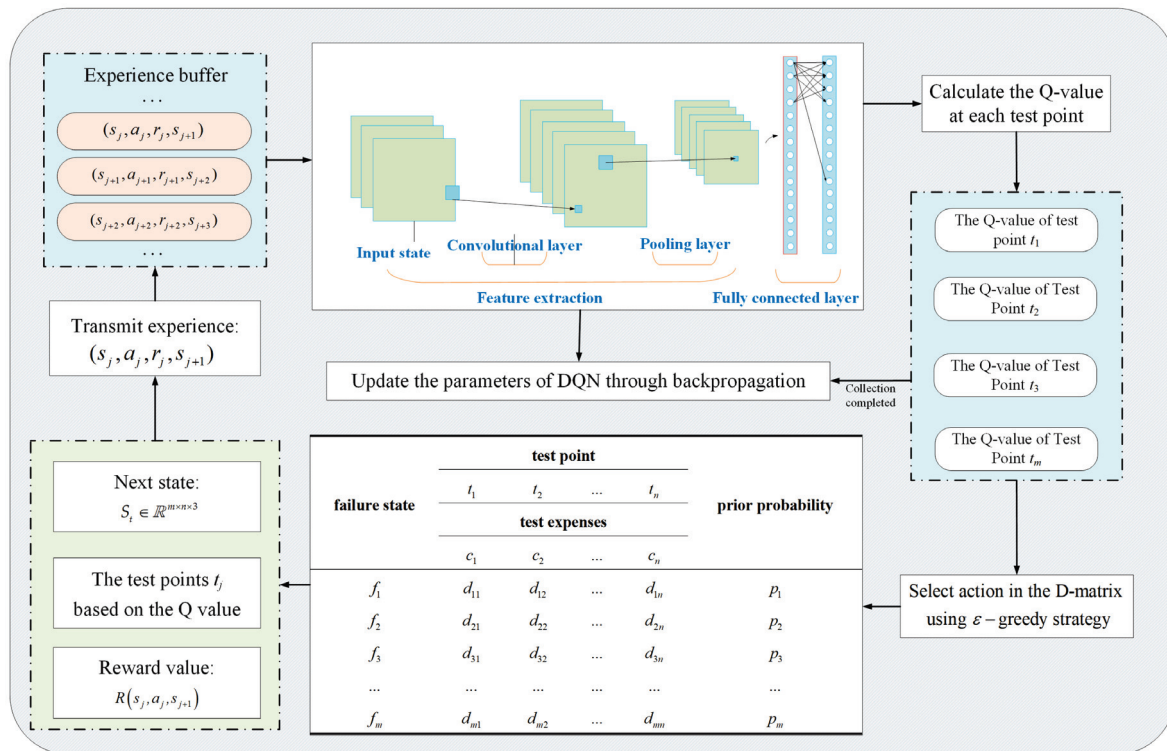


Figure 3. Q-network structure.

During the DQN training process, the agent can adopt the ϵ -greedy strategy to select actions, thereby ensuring the comprehensiveness of sample coverage. In other words, the agent will select the action with the highest Q-value with a probability of $1 - \epsilon$ to exploit the learned policy, and explore other random actions with a probability of ϵ .

Step 2. Experience collection

The agent interacts with the MVD matrix by selecting actions based on the output of the fully connected layers of the policy network. After performing an action, the environment provides a reward and transitions to the next state. The agent stores this experience (s_j, a_j, r_j, s_{j+1}) , including the current state, the executed action, the obtained reward, and the next state, in the buffer pool. If the buffer pool is full, new experiences will overwrite the earliest stored ones.

Step 3. Sampling and calculation of Q-value

A batch of experiences is randomly sampled from the buffer pool. Assume that the batch size is N . These samples are obtained through uniform random sampling from the replay buffer, and each sample contains a complete experience (s_j, a_j, r_j, s_{j+1}) . For each sampled experience, the Q-value y_j is calculated. If the next state s_{j+1} is a terminal state, then $y_j = r_j$; otherwise, the Q-value calculation formula is

$$y_j = r_j + \gamma \max_a Q_2(s_{j+1}, a; \theta^-), \quad (12)$$

where y_j is the target Q-value of the j -th sampling experience; r_j is the immediate reward obtained by the agent when it transitions from state s_j to state s_{j+1} after taking action a_j in the j -th sampling experience; γ is the discount factor; $Q_2(s_{j+1}, a; \theta^-)$ is the Q-value estimate for taking action a in state s_{j+1} , calculated through the target network; θ^- is the parameters of the target network.

Step 4. Loss calculation and backpropagation

For each sampled empirical sample, the current state s_j is input into the Q-network to obtain the predicted Q-value corresponding to the executed action a_j .

The loss function adopts the mean square error (MSE), and its expression is shown as

$$L(\theta) = \frac{1}{N} \sum_j (y_j - Q_1(s_j, a_j; \theta))^2. \quad (13)$$

This loss function measures the difference between the predicted Q-value and the target Q-value. The training objective is to minimize this loss function, so that the Q-network can better approximate the target Q-value.

Based on the calculated loss, the parameters θ of the policy network are updated through the backpropagation algorithm. The backpropagation algorithm starts from the loss function and calculates the gradients along the hierarchical structure of the Q-network in reverse. The parameters of the network are updated according to the gradients and the learning rate. The gradient update of the parameters is shown as

$$\theta \leftarrow \theta - lr \cdot \nabla_{\theta} L(\theta), \quad (14)$$

where lr is the learning rate.

Step 5. Update of target network

After a certain number of steps, the target network needs to be checked and updated. If the update conditions are met, the parameters of the policy network are copied to the target network:

$$\theta^- \leftarrow \theta. \quad (15)$$

The parameters of the target network will keep up with the update pace of the policy network and maintain the relative stability.

3.2. The Optimization of the DQN Model—PE-DDQN

As the system becomes increasingly complex, the high-dimensional MVD matrix exacerbates the overestimation issue in DQN caused by selecting the maximum Q-value during training and error accumulation. The large five-tuple matrix structure increases the diversity of experience samples, while traditional experience replay cannot prioritize important samples. To address these problems, we combine DDQN and PER to construct the PE-DDQN model to adapt to the high-dimensional MVD matrix environment.

As the system becomes increasingly complex, the high-dimensional MVD matrix exacerbates the overestimation issue in DQN caused by selecting the maximum Q-value during training and error accumulation. The large five-tuple matrix structure increases the diversity of experience samples, while traditional experience replay cannot prioritize important samples. To address these problems, we combine DDQN and PER to construct the PE-DDQN model to adapt to the high-dimensional MVD matrix environment. The overestimation and sample-redundancy problems noted above reflect a broader observation that, in high-dimensional and continuous settings, reinforcement learning must contend with sparse, noisy experience and with the tension between model-free and model-based updating; stable, reward-relevant state representations have been argued to be a prerequisite for reliable learning under these conditions [42]. In the present method, the three-channel

encoding together with prioritized experience replay and the decoupled Double-DQN targets serve this role, suppressing the overestimation and sample-redundancy effects that the high-dimensional MVD matrix would otherwise amplify.

3.2.1. DDQN

The overestimation problem exists in the aggressive Q-learning algorithm. DDQN is mainly designed to address the suboptimal strategy issue caused by the overestimation of action values in Q-learning. It separates action selection from action evaluation. When choosing the next action a^* , the Q_1 function is used for selection, which is shown in

$$a^* = \operatorname{argmax}_a Q_1(s_{j+1}, a). \quad (16)$$

When calculating the target value, another function Q_2 is used to evaluate the value of this action. The formula for updating the target value becomes

$$Q_1(s_j, a_j) \leftarrow Q_1(s_j, a_j) + \alpha[r_j + \gamma Q_2(s_{j+1}, a^*) - Q_1(s_j, a_j)]. \quad (17)$$

This operation can reduce the overestimation phenomenon caused by always choosing the maximum estimate value. By regularly exchanging the parameters of Q_1 and Q_2 , the two Q-networks can both participate in action selection and evaluation, ensuring fairness and accuracy.

3.2.2. PER Mechanism

PER breaks the uniform sampling of traditional experience replay, and prioritizes learning important samples based on temporal difference error (TD-error). For an experience sample (s_j, a_j, r_j, s_{j+1}) , the formula for calculating the TD-error is

$$\delta = r_j + \gamma Q_2(s_{j+1}, \operatorname{argmax}_{a_{j+1}} Q_1(s_{j+1}, a_{j+1}, \theta); \theta^-) - Q_1(s_j, a_j; \theta). \quad (18)$$

The larger the TD-error is, the greater the gap between the predicted value of the current sample and the target value will be. This means that this sample is more likely to contain important information, such as it might be a sample that can isolate multiple test points, or a sample located at a critical decision point.

After calculating the TD-error for each sample, a priority function $p = f(\delta)$ is used to determine the priority of the samples. The specific approach is to set $p = |\delta| + \epsilon$, where ϵ is a small positive number used to ensure that all samples have non-zero priorities. This ensures that even samples with a TD-error of 0 have a certain chance of being selected. In the experience replay buffer, in addition to storing the experience samples (s_j, a_j, r_j, s_{j+1}) , the priority p of each sample also needs to be stored. This can be achieved using a priority queue or an array with priority information.

Next, sampling is carried out based on the priority of the samples. One common sampling method is based on probability distribution, and this probability distribution is related to the priority of the samples. Specifically, the probability formula for selecting sample i is shown as

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \quad (19)$$

where $P(i)$ is the priority of sample i ; α is the PER prioritization exponent hyperparameter that determines the degree of influence on priority control. When $\alpha = 0$, the sampling becomes uniform sampling; the larger α is, the greater the probability that the samples with higher priority will be selected.

To avoid always selecting the sample with the highest priority, importance sampling weight w_i is applied to adjust the contribution of the sampled samples to the training, and its calculation formula is shown as

$$w_i = \frac{(N \times P(i))^{-\beta}}{\max_j w_j}, \quad (20)$$

where N is the total number of samples in the experience buffer zone; β is the PER importance sampling hyperparameter used to adjust the degree of adjustment of the importance sampling weights.

3.3. Algorithm Implementation Steps

Based on the traditional DQN, DDQN and PER are introduced to construct a PE-DDQN model for testability strategy optimization. The specific implementation of the algorithm process of PE-DDQN is shown in Figure 4. The steps for establishing the PE-DDQN can be summarized as five steps as follows.

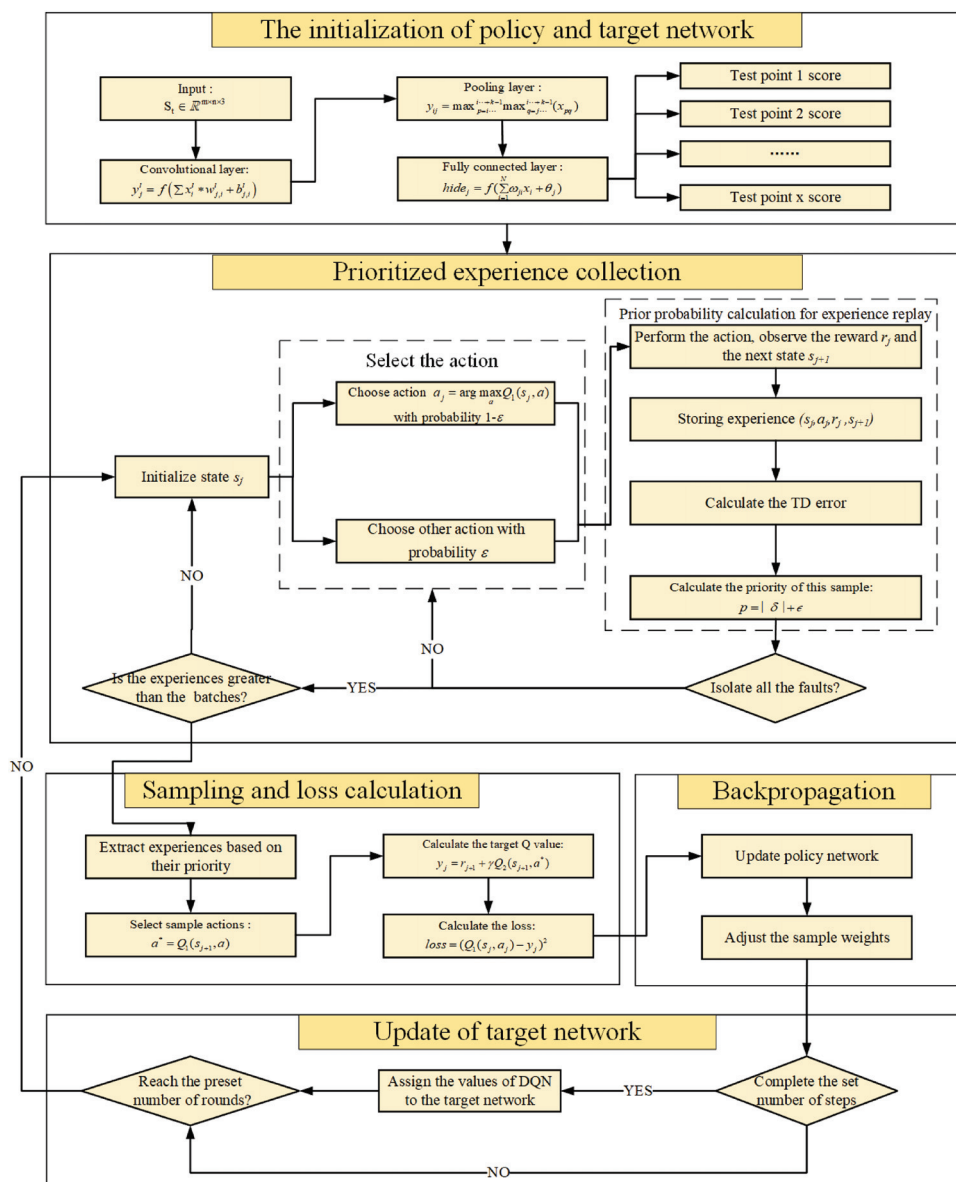


Figure 4. The flowchart of PE-DDQN.

- (1) The initialization of policy network and target network. Firstly, the parameters of the DQN and the target network need to be initialized separately. The network structure adopts a convolutional neural network.
- (2) Priority experience collection. The agent selects test points through a greedy strategy and obtains the next state and the reward for that step from the MVD matrix to obtain experience samples and store it in the buffer pool. This process continues until all faults are isolated and the termination state is reached. At the same time, for each sample, the TD error is calculated, and a priority is assigned.
- (3) Sampling and loss calculation. When the number of samples in the pool is greater than the batch size, the algorithm will select samples according to priority. The agent uses the policy network to select actions and calculate the current Q-value, and uses the target network to calculate the target Q-value.
- (4) Backpropagation. The policy network parameters are updated through backpropagation, and the updated parameters consist of two parts. One part consists of the parameters of the convolutional neural network in DQN, and the second part is the weight parameters of the PER pool.
- (5) Update of the target network. After a certain number of training sessions, the parameters of the DQN will be assigned to the target network for the update of the target network.

After completing the steps, the agent will return to the second step to begin collecting priority experience samples and initiate the next round of training. The algorithm will repeat the above steps until the network converges or the preset number of training iterations is reached.

4. Experimental Verification

Before delving into the specific case studies and algorithmic comparisons, the network architectures and overall training configurations are established by adjusting parameters. The detailed structural parameters of the DQN utilized for state feature extraction are summarized in Table 1.

Table 1. The structure of DQN.

Layer Type	Size	Stride	Number of Convolutional Kernels	Explanation
Input layer	$m \times n \times 3$	--	--	Receives the 3-channel 2D state tensor.
Convolutional layer	3×3	1×1	32	Extracts initial coupling features.
Pooling layer	2×2	2×2	--	Reduces dimensions, retains salient features.
Convolutional layer	3×3	1×1	64	Expands receptive field for higher-order features.
Pooling layer	2×2	2×2	--	Compresses spatial redundancy.
Flatten layer	--	--	--	Flattens 2D maps into 1D.
Fully connected layer	512	--	--	Fuses all features globally.
Fully connected layer	n	--	--	Outputs Q-values for n test points.

Furthermore, to ensure strict methodological reproducibility, all vital hyperparameters involved in the training phase of the PE-DDQN agent are explicitly defined in Table 2. This set of hyperparameters can enable the algorithm to balance both training stability and training efficiency during the subsequent training process.

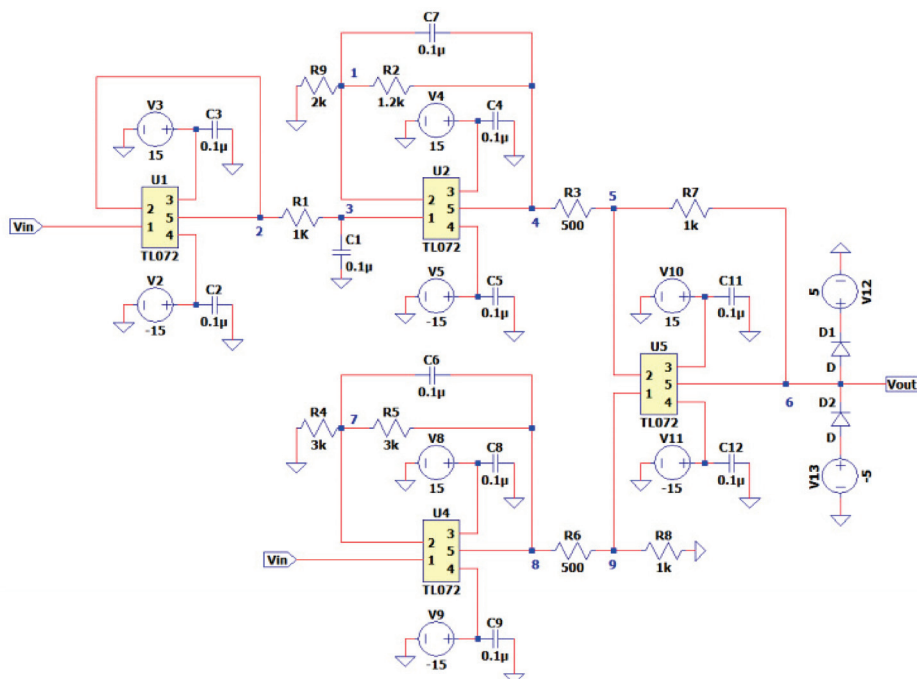
Table 2. Hyperparameters for training.

Parameter	Symbol	Value	Description
Learning rate	lr	0.001	Step size for Adam optimizer
Discount factor	γ	0.99	Importance of future rewards
Replay buffer size	-	20,000	Max experience capacity
Mini-batch size	-	64	Samples per network update
PER prioritization exponent	α	0.6	Degree of prioritized sampling
PER importance sampling	β	0.4~1.0	Importance sampling weight
Target-update interval	-	500 steps	Target network sync frequency

4.1. Experimental Case 1

4.1.1. MVD-Matrix Modeling

The first research object is a sampling circuit composed of an amplification and subtraction circuit, which is shown in Figure 5. It is an important component of a digital signal system, which can provide a reliable data source for subsequent digital signal processing and enhance the flexibility and multi-band adaptability of the system. A total of 14 fault conditions (open/short circuit of resistors and short circuit of capacitors) are defined, and together with the normal state, there are 15 working conditions, and 9 test points are set.

**Figure 5.** Sampling circuit.

The circuit voltage excitation is set as a sinusoidal AC voltage source with an amplitude of 4 V and a frequency of 1 kHz. The transient analysis voltage signal is used until stabilization. Based on the actual experiment and exported data, the set simulation time is observed to be 10 ms, and the maximum time step is 1 ms. The experiment recorded the effective values of each test point under different fault conditions. Based on the 0.3 V voltage drop threshold, the MVD matrix of the sampling circuit is constructed by fuzzy integer encoding, as shown in Table 3. The redundant test point T3 is eliminated because its multi-valued fault response column is completely identical to test point T1, offering no additional fault isolation information.

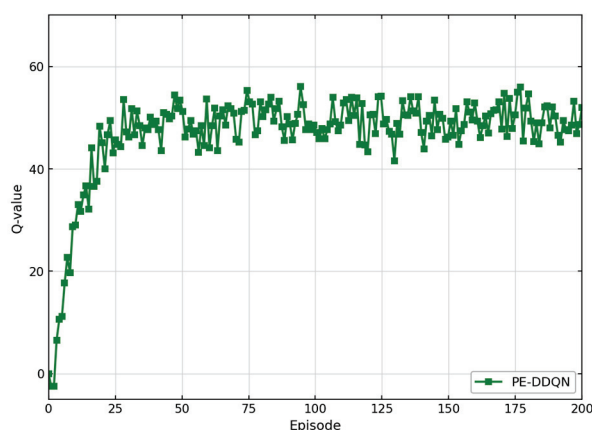
Table 3. MVD matrix of sampling circuit.

FT	T1	T2	T4	T5	T6	T7	T8	T9
F0 (normal)	1	0	2	4	4	1	1	2
F1 (R1 open)	0	1	0	1	8	1	1	2
F2 (R1 short)	2	1	3	4	0	1	1	2
F3 (R2 open)	1	0	4	3	6	0	1	2
F4 (R2 short)	1	1	1	4	3	1	1	2
F5 (R3 open)	1	1	2	4	2	1	1	2
F6 (R3 short)	1	1	2	7	9	1	1	2
F7 (R5 open)	1	0	2	5	5	1	1	2
F8 (R5 short)	1	0	2	2	6	1	0	1
F9 (R6 open)	1	0	2	0	8	1	1	0
F10 (R6 short)	1	1	2	7	7	1	1	3
F11 (R7 open)	1	0	2	7	9	1	1	2
F12 (R7 short)	1	0	2	4	2	1	1	2
F13 (C6 short)	1	1	2	6	8	1	2	3
F14 (C7 short)	1	1	3	4	1	1	1	2

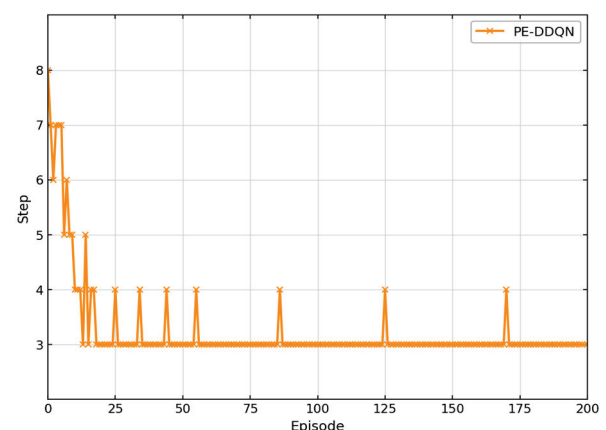
4.1.2. Optimization of Test Sequence Length

For this simulation circuit, the failure rates of all components are approximately equal, and the difficulty of conducting tests at each test point is comparable. Therefore, our objective is to take the fewest number of steps to quickly isolate all faults. With the aim of minimizing the test sequence length, the corresponding reward function is selected. The PE-DDQN testability model is constructed and trained according to the steps described in the third subsection. The maximum number of iterations is 200.

The average Q-value during the training process is shown in Figure 6a. When the average Q-value rises to a certain extent, it enters a state of stable small fluctuations. This indicates that the algorithm has achieved a certain degree of convergence and has found a relatively stable strategy. The small fluctuations are caused by the setting of the exploration rate. However, overall, the average Q-value remains within a relatively stable range, indicating that the algorithm has good stability. This stability enables the agent to make relatively consistent decisions when facing similar situations, improving the reliability of the algorithm in practical applications.



(a)



(b)

Figure 6. The training process of PE-DDQN: (a) average Q-value; (b) required steps.

The steps required for the agent to reach the destination are shown in Figure 6b. Initially, the required test sequence length fluctuates considerably due to the agent's explo-

ration and unfamiliarity with the environment, leading to inefficient actions and longer sequences. As training proceeds, the agent learns and optimizes its strategy, so the sequence length decreases rapidly. After stabilizing at a low level, minor occasional fluctuations demonstrate the algorithm's adaptability to environmental uncertainties.

Subsequently, the initial state is selected and sent to the trained model. The test point with the highest Q-value is taken and updated to the next state. This step is repeated until all faults are isolated. The final output test sequence is [T5, T6, T2]. According to Table 3, the diagnostic tree can be plotted based on the output test sequence.

At the initial state, all faults are not distinguished, meaning all faults are in a fuzzy group. Firstly, the agent selects test point T5, which divides the fault group into eight parts: {F9}, {F1}, {F8}, {F3}, {F0, F2, F4, F5, F12, F14}, {F7}, {F13}, {F6, F10, F11}. Then, the test point T6 in the current state is selected, which divides the two incomplete isolated fuzzy groups into {F0}, {F2}, {F4}, {F5, F12}, {F14}, {F6, F11}, {F10}. These fuzzy groups, along with the faults F9, F1, F8, F3, F7, F13, isolated in the previous round, indicate that there are currently 11 isolated faults and two incomplete isolated fuzzy groups of faults. Finally, the agent selects test point T2 for the test. At this point, all faults will be completely isolated. According to the above logic, the drawn diagnostic tree is shown in Figure 7. In the figure, the green boxes represent the selected test points, and the gray circles represent the faulty fuzzy groups.

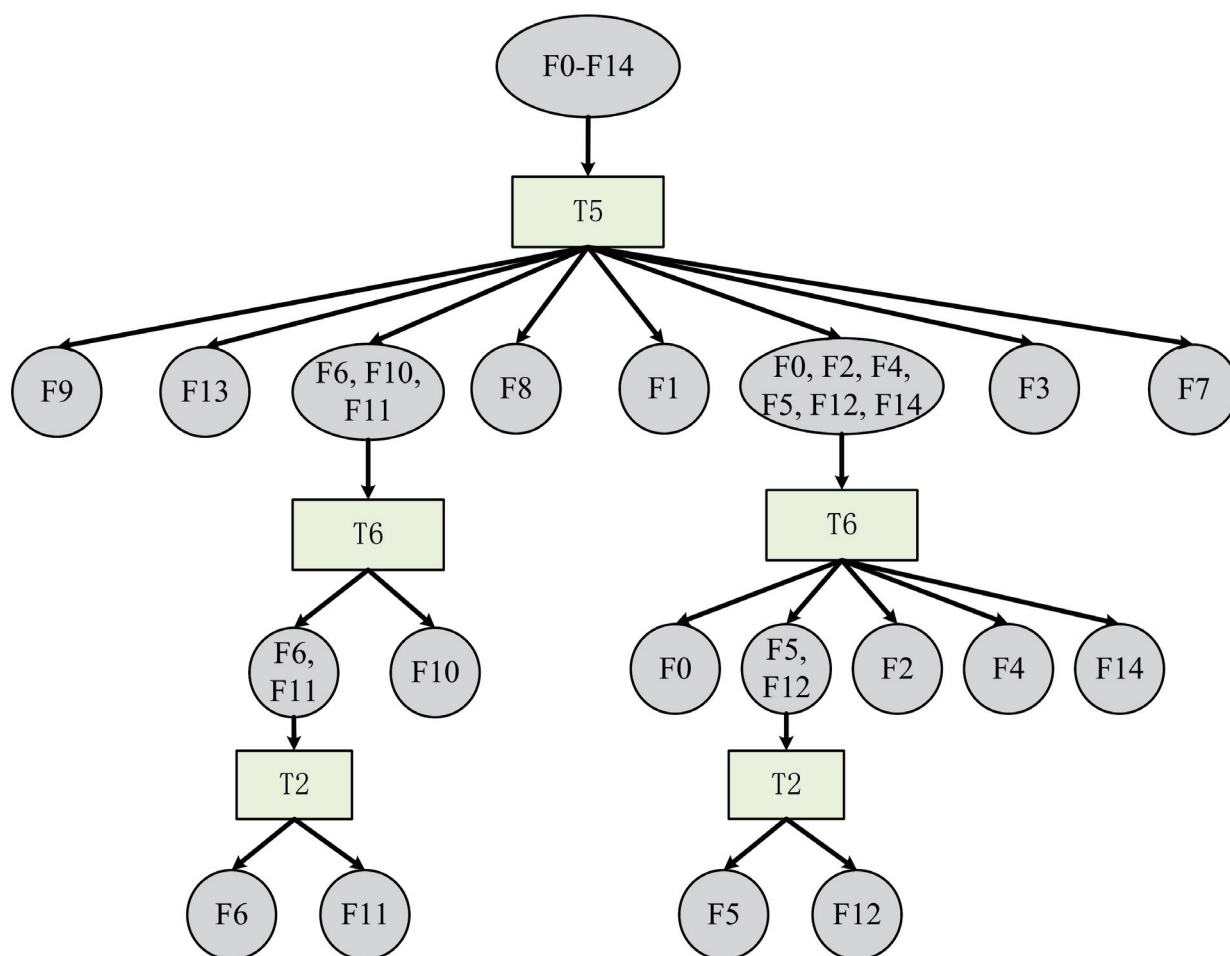


Figure 7. Diagnostic tree based on test sequence length.

By adjusting the initial seed during the model's operation, five independent model training sessions are conducted, and the final test sequence is the output. The length output by the five experiments is all three, and the results are shown in Table 4. From the output

results, it can be seen that although the test sequence and sorting methods vary each time, they can each successfully isolate every fault through three testability tests. This indicates that there is more than one sequence that can isolate all the faults with the minimum number of diagnostic attempts, but they are all the shortest test sequences.

Table 4. The sequences output from the five experiments.

Number of Experiments	1	2	3	4	5
Output sequence	T5→T6→T2	T6→T2→T5	T6→T2→T5	T6→T2→T9	T9→T6→T2

4.2. Experimental Case 2

4.2.1. MVD-Matrix Modeling

The second case study focuses on the MIRFS, which is a shared electronic platform that relies on radio frequency integration, software integration, and aperture integration technologies. Currently, it has been widely applied in military electronic countermeasures and civil aircraft communication. The structure diagram of MIRFS studied in this paper is shown in Figure 8. It replaces the complex discrete antenna apertures with a broadband distributed multi-functional aperture, and realizes resource sharing and function reconstruction by using scheduling control algorithms and software programming. It mainly consists of three core modules: radio frequency (RF) processing, signal processing, and integrated management. These modules work in synergy to implement the functions from RF signal reception and processing to integrated management and output. Meanwhile, it interacts with the avionics bus and is integrated into the overall avionics system.

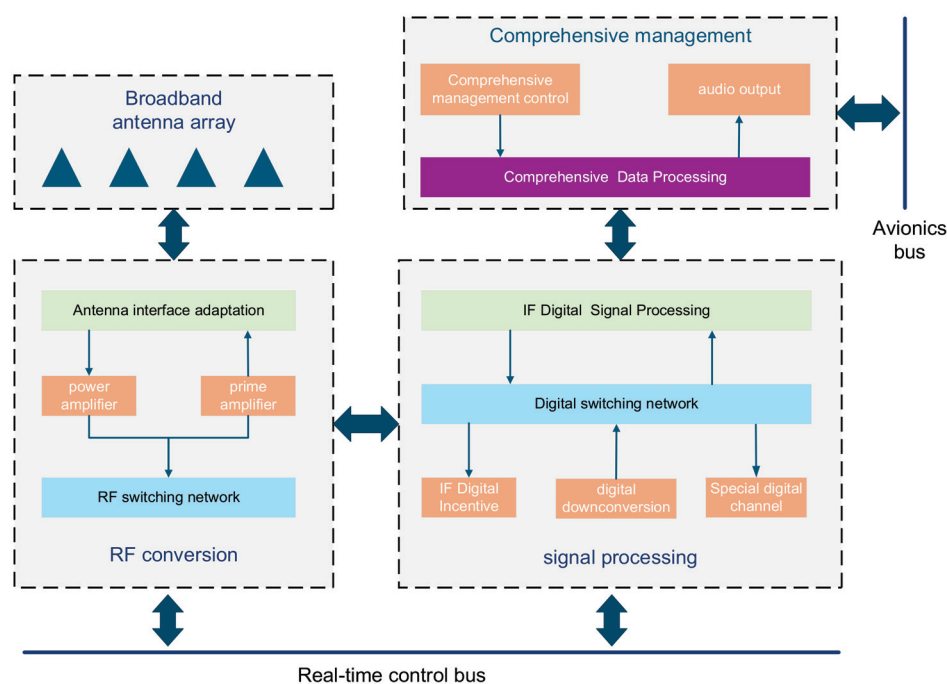


Figure 8. The structure of MIRFS.

Based on the system structure composition of the aforementioned MIRFS, the MVD matrix model of MIRFS is shown in Table 5. This fault information covers the normal state of the system and 20 typical faults, and also provides the occurrence probabilities of these faults. The occurrence probabilities are derived based on the operational patterns of the system in actual engineering. The test information contains 10 test points derived from the MIRFS architecture, with their respective execution costs specified. The test items

are: T1 (antenna Standing Wave Ratio), T2 (Power amplifier (PA) output power), T3 (RF switch time), T4 (Intermediate frequency (IF) amplitude), T5 (down-conversion gain), T6 (digital channel), T7 (data processing), T8 (control bus load), T9 (avionics bus rate), and T10 (system connectivity). The execution costs are set differently based on the complexity and operational difficulty of the detection modules. The values in the matrix represent the detection response values of each fault at the corresponding test points. Test points with insignificant fault differences will be grouped into the same fuzzy category.

Table 5. MVD matrix of MIRFS.

Fault (Probability)	Fault Message	T1 (1.0)	T2 (1.5)	T3 (2.0)	T4 (1.8)	T5 (2.0)	T6 (1.0)	T7 (1.6)	T8 (0.8)	T9 (1.1)	T10 (2.6)
F0 (0.8500)	No fault	1	1	1	1	1	1	1	1	1	1
F1 (0.0050)	Antenna short circuit	0	0	0	0	0	0	0	1	1	2
F2 (0.0090)	RF switch jamming	3	2	4	3	4	4	1	1	0	0
F3 (0.0050)	Main PA gain loss	1	1	0	0	0	0	0	1	1	2
F4 (0.0080)	Power PA overload	2	3	6	5	6	6	1	1	2	0
F5 (0.0120)	Antenna impedance mismatch	1	1	1	1	1	2	1	1	5	0
F6 (0.0070)	IF signal failure	0	0	3	2	3	3	1	1	8	0
F7 (0.0080)	Conversion gain drift	3	2	5	4	5	5	1	1	4	0
F8 (0.0060)	RF switch input fault	3	2	0	0	0	0	1	1	7	0
F9 (0.0090)	Digital channel overflow	3	2	6	6	7	7	1	1	6	0
F10 (0.0060)	Signal sampling fault	3	2	2	0	0	0	1	1	7	0
F11 (0.0070)	Switch cross fault	3	2	3	3	4	2	1	1	4	0
F12 (0.0080)	IF filter abnormality	3	2	3	2	0	8	5	1	8	0
F13 (0.0090)	Data processing loss	3	2	3	2	2	8	4	1	8	0
F14 (0.0070)	Management command delay	3	2	3	3	4	4	1	1	0	0
F15 (0.0050)	Audio distortion	3	2	3	3	4	4	2	1	8	0
F16 (0.0080)	Switch output fault	3	2	3	3	4	4	3	1	8	0
F17 (0.0050)	IF sampling error	3	2	3	3	4	4	2	2	3	0
F18 (0.0080)	Management comm. interrupt	3	2	3	3	4	4	0	0	1	0
F19 (0.0050)	Control bus error	1	3	2	3	3	4	0	2	1	0
F20 (0.0130)	Bus communication fault	2	2	3	4	2	4	1	0	1	0

4.2.2. Optimization of Test Cost

For the MIRFS system, the complexity and operational difficulty of its test points are different, and the failure probabilities of each component also vary. Therefore, we consider using a low testing cost to isolate all the faults. Similarly, with the aim of minimizing the test cost, the corresponding reward function is selected. The PE-DDQN testability model is constructed and trained according to the steps described in the third subsection, whose process is shown in Figure 9. Compared with the reward function based on the test sequence length, the reward function based on the test cost increases the reward for isolating faults, and thus, the overall Q-value during the training process is higher than that of the test sequence length. Meanwhile, the test cost output by the model also tends to converge.

By inputting the initial state vector into the trained model, the test sequence corresponding to the minimum test cost is output. The mean and standard deviation of the results from five independent repeated experiments with random seeds are 1.2404 ± 0.0040 . The lowest testing cost among the five experiments is 1.2370, and the costs for each fault are shown in Table 6. The final output of the test point set includes [T6, T1, T9, T7, T3]. Unlike the optimization of test sequence length, this is merely a possible set of measurement points that might be adopted during the testing process, rather than a fixed testing sequence. The actual testing process will dynamically adjust the test points based on the results of each diagnostic step, thereby forming a dynamic diagnostic tree. The execution begins with

T6, which serves as a highly discriminative root node, isolating faults F0, F4, F6, F7, F9 and F12, F13 (via further branches). Following the diagnostic tree, the agent dynamically selects subsequent tests based on the specific ambiguity group remaining. For example, for the subgroup [1,3,8,10], the agent performs T1 to isolate F1 and F3, and further utilizes T3 to distinguish F8 and F10. Notably, for the complex subgroup [2,14–20], the agent adopts T9, followed by T1, T3 and T7 in a branching manner, ensuring that each fault is isolated through the shortest path relative to its individual probability-weighted cost. The diagnostic tree is shown in Figure 10. Enumeration verification confirms that this test yields the lowest test cost.

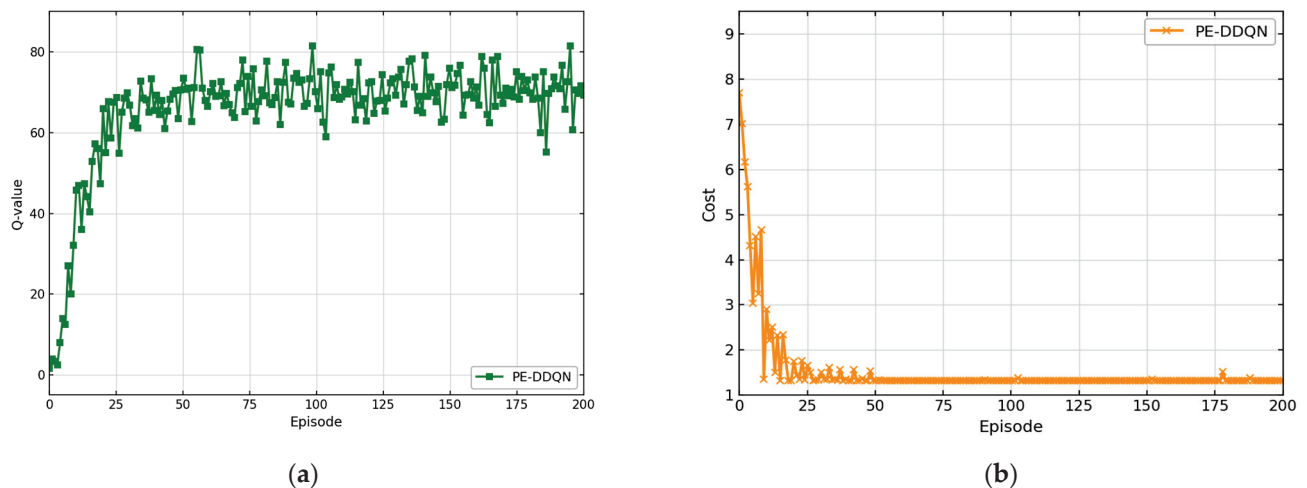


Figure 9. The training process of PE-DDQN: (a) average Q-value; (b) test cost.

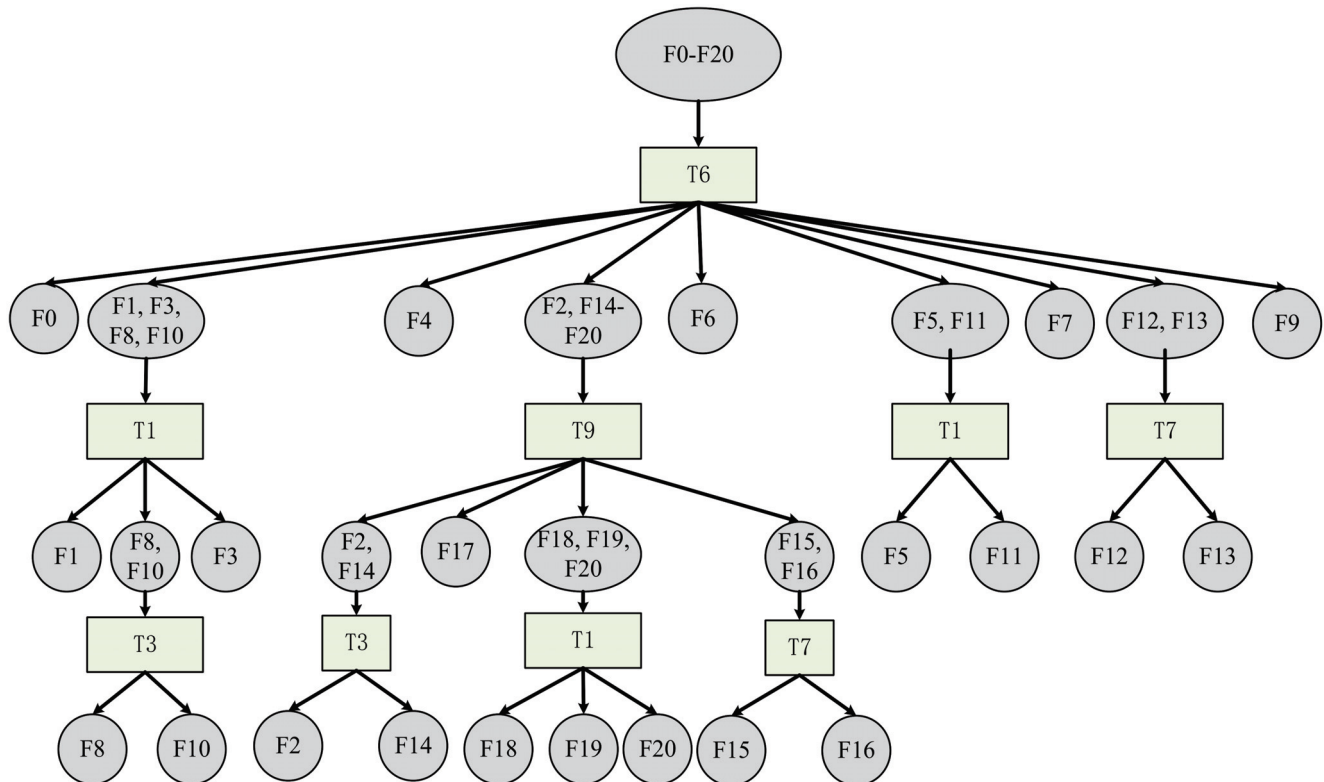


Figure 10. Diagnostic tree based on test cost.

Table 6. Test cost table for each fault.

No.	Test Points	Cost	No.	Test Points	Cost	No.	Test Points	Cost
F0	T6	0.85000	F7	T6	0.00800	F14	T6→T9→T3	0.02870
F1	T6→T1	0.01000	F8	T6→T1→T3	0.02400	F15	T6→T9→T7	0.01850
F2	T6→T9→T3	0.03690	F9	T6	0.00900	F16	T6→T9→T7	0.02960
F3	T6→T1	0.01000	F10	T6→T1→T3	0.02400	F17	T6→T9	0.01050
F4	T6	0.00800	F11	T6→T1	0.01400	F18	T6→T9→T1	0.02480
F5	T6→T1	0.02400	F12	T6→T7	0.02080	F19	T6→T9→T1	0.01550
F6	T6	0.00700	F13	T6→T7	0.02340	F20	T6→T9→T1	0.04030

4.2.3. The Comparison of the Two Reward Mechanisms

In order to compare the characteristics of the two reward mechanisms, the length of the test sequence for MVD matrix of MIRFS is optimized. Under the test sequence length optimization reward function, the output test sequence is T9, T3, T1 and T7, and the total test cost corresponding to the output sequence is 4.948. The comparison of the outputs of the two reward mechanisms is shown in Table 7. The cost differences between the two mainly lie in two aspects. The algorithm based on the test cost optimization reward mechanism considers how to isolate the most likely occurring fault F0 with the smallest test cost combination as soon as possible, while the algorithm based on the test sequence length optimization reward mechanism considers how to isolate all faults more quickly. Although the number of test points has increased, it has, to some extent, reduced the test cost and enabled the test to be conducted in a more economical manner.

Table 7. Comparison of two reward mechanisms.

Reward Mechanism	Test Sequence (Set)	Sequence Length	Test Cost	Core Difference
Test sequence length	T9→T3→T1→T7	4	4.948	Fewer steps and faster speed
Test cost	[T6, T1, T9, T7, T3]	5	1.2370	More economical

4.3. Contrastive Analysis

4.3.1. Comparison with Traditional Algorithms

In order to study the performance differences in the proposed algorithm compared to the traditional testability strategy optimization methods, as well as its outstanding superiority in high-dimensional matrix, this paper compares the optimization performance of the proposed PE-DDQN algorithm with that of six traditional optimization algorithms under various dimensional MVD matrix, including ant colony (AC), growing, rollout, info-entropy greedy (IG), genetic algorithm (GA), and particle swarm optimization (PSO). The MVD matrix dimension used in the contrastive experiment increases in dimension, ranging from 20 to 200. For each matrix dimension, 30 repeated experiments are conducted using seven different algorithms, ensuring the reliability of the experimental results. Elements within the matrix are randomly generated, and a certain change is added in each repeated experiment. Since the optimal solution for the shortest sequence length often has multiple possibilities, this optimization objective is relatively simple and cannot fully reflect the superiority or inferiority of the algorithm. Therefore, the complex and representative test cost is selected as the optimization objective.

Table 8 presents the average test cost and standard deviation obtained from 30 experiments of seven algorithms across various MVD matrix dimensions. It can be observed from Table 8 that some heuristic algorithms, such as GA and PSO, maintain a relatively good ability to search for the optimal test cost when the matrix dimension is relatively small. The average value of the results from multiple experiments is not significantly different

from that of PE-DDQN. However, as the dimension of the matrix increases, the average test cost of all traditional optimization algorithms gradually exceeds that of PE-DDQN during the process of continuous increase, and thus falls into a local optimum. In contrast, the proposed PE-DDQN algorithm maintains the lowest average test cost across all tested high-dimensional scenarios, demonstrating stable and superior cost optimization capability. This is because these heuristic methods require the analysis and calculation of a large amount of data, involving numerous variables and complex logical relationships. As the scale of the system continues to expand and its complexity deepens, the computational load will show an exponential growth trend, making it impossible to find the optimal solution. Meanwhile, the relatively small standard deviation of the PE-DDQN algorithm as a whole also indicates that its stability is higher compared to other algorithms, and it has broader application value.

Table 8. Comparison of average test cost and standard deviation for 30 trials.

Dim	PE-DDQN	AC	Growing	Rollout	IG	GA	PSO
20	2.155 ± 0.0029	2.165 ± 0.0017	2.184 ± 0.0020	2.167 ± 0.0021	2.213 ± 0.0032	2.163 ± 0.0020	2.165 ± 0.0018
40	2.215 ± 0.0033	2.235 ± 0.0023	2.275 ± 0.0024	2.242 ± 0.0028	2.246 ± 0.0016	2.229 ± 0.0024	2.226 ± 0.0020
60	2.236 ± 0.0032	2.329 ± 0.0021	2.395 ± 0.0029	2.298 ± 0.0036	2.444 ± 0.0026	2.260 ± 0.0030	2.273 ± 0.0026
80	2.285 ± 0.0024	2.316 ± 0.0043	2.449 ± 0.0025	2.388 ± 0.0034	2.516 ± 0.0022	2.323 ± 0.0025	2.303 ± 0.0033
100	2.360 ± 0.0032	2.426 ± 0.0033	2.554 ± 0.0027	2.474 ± 0.0048	2.634 ± 0.0041	2.457 ± 0.0035	2.445 ± 0.0029
120	2.507 ± 0.0024	2.685 ± 0.0028	2.853 ± 0.0027	2.606 ± 0.0062	2.973 ± 0.0021	2.628 ± 0.0036	2.627 ± 0.0035
140	2.590 ± 0.0026	2.798 ± 0.0050	3.014 ± 0.0022	2.785 ± 0.0058	3.158 ± 0.0032	2.769 ± 0.0061	2.732 ± 0.0038
160	2.683 ± 0.0021	2.887 ± 0.0045	3.061 ± 0.0028	2.886 ± 0.0052	3.177 ± 0.0020	2.870 ± 0.0049	2.818 ± 0.0050
180	2.824 ± 0.0029	2.967 ± 0.0046	3.070 ± 0.0021	2.968 ± 0.0048	3.191 ± 0.0045	2.964 ± 0.0039	2.911 ± 0.0072
200	2.941 ± 0.0022	3.017 ± 0.0055	3.084 ± 0.0023	3.069 ± 0.0059	3.192 ± 0.0024	3.081 ± 0.0078	3.027 ± 0.0031

In order to further quantitatively verify the statistical reliability of the performance advantages of the PE-DDQN algorithm under various conditions of dimensions, a two-way analysis of variance (ANOVA) is conducted based on dimensions and algorithms. Taking into account that there might be differences in variance among different datasets, the ANOVA with HC3 robust standard errors is employed for the correctional test. The ANOVA analysis results for the main effects of the algorithms are $F = 11001$ and $p < 0.001$. This indicates that, in general, these eight algorithms have fundamental performance differences in terms of overall cost optimization capabilities. The analysis results of the interaction effect are $F = 586.2$ and $p < 0.001$, which reveals a highly significant interaction effect between the algorithm and matrix dimension factors.

Subsequently, we conduct a one-way ANOVA and pairwise post hoc test in different dimensions, with the algorithm serving as the variable. Considering that the variances are different among the groups, we employ the Welch's ANOVA and Games–Howell post hoc tests. In the case of a 200-dimensional matrix, the one-way ANOVA result is $F = 1018$ and $p < 0.001$, demonstrating significant overall differences in optimization performance among all compared algorithms, which justifies carrying out the subsequent experiments. Based on this, an independent pairwise post hoc test is conducted between each algorithm, with corresponding p -value calculated to quantify performance differences and statistical significance via a comparison matrix, which is shown in Table 9. The elements in the matrix represent the p -value corresponding to the two algorithms. The results show that all p -values of PE-DDQN against other algorithms are less than 0.001, indicating that PE-DDQN achieves significantly lower test cost than all comparative algorithms with extremely significant statistical advantages. Among them, the p -value corresponding to the multi-valued rollout algorithm is the smallest, meaning that the performance gap between PE-DDQN and the rollout algorithm is the most prominent under high-dimensional conditions. In addition, non-significant pairwise comparisons further show the grouping

of significance letters, where algorithms sharing identical letters exhibit no statistically significant performance difference, where $p \geq 0.05$. Figure 11 shows the error range of costs in the 30 experiments for each algorithm and the grouping situation of algorithm performance, intuitively reflecting the stability and superiority of the PE-DDQN algorithm in terms of test cost. Based on the statistical significance of the letter markings, the seven algorithms were divided into four distinct groups. Among them, PE-DDQN is classified as group a, indicating its significant advantages and stability compared to other algorithms. AC and PSO are grouped as group b. Growing, GA, and Rollout are grouped as group c. IG is classified as group d alone, which indicates that the performance of this algorithm is the worst.

Table 9. Pairwise p -value matrix (Dimension = 200).

Algorithm	PE-DDQN	AC	Growing	Rollout	IG	GA	PSO
PE-DDQN	—	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
AC	<0.001	—	<0.001	<0.001	<0.001	<0.001	0.7114
Growing	<0.001	<0.001	—	0.2892	<0.001	0.9996	<0.001
Rollout	<0.001	<0.001	0.2892	—	<0.001	0.9148	<0.001
IG	<0.001	<0.001	<0.001	<0.001	—	<0.001	<0.001
GA	<0.001	<0.001	0.9996	0.9148	<0.001	—	<0.001
PSO	<0.001	0.7114	<0.001	<0.001	<0.001	<0.001	—

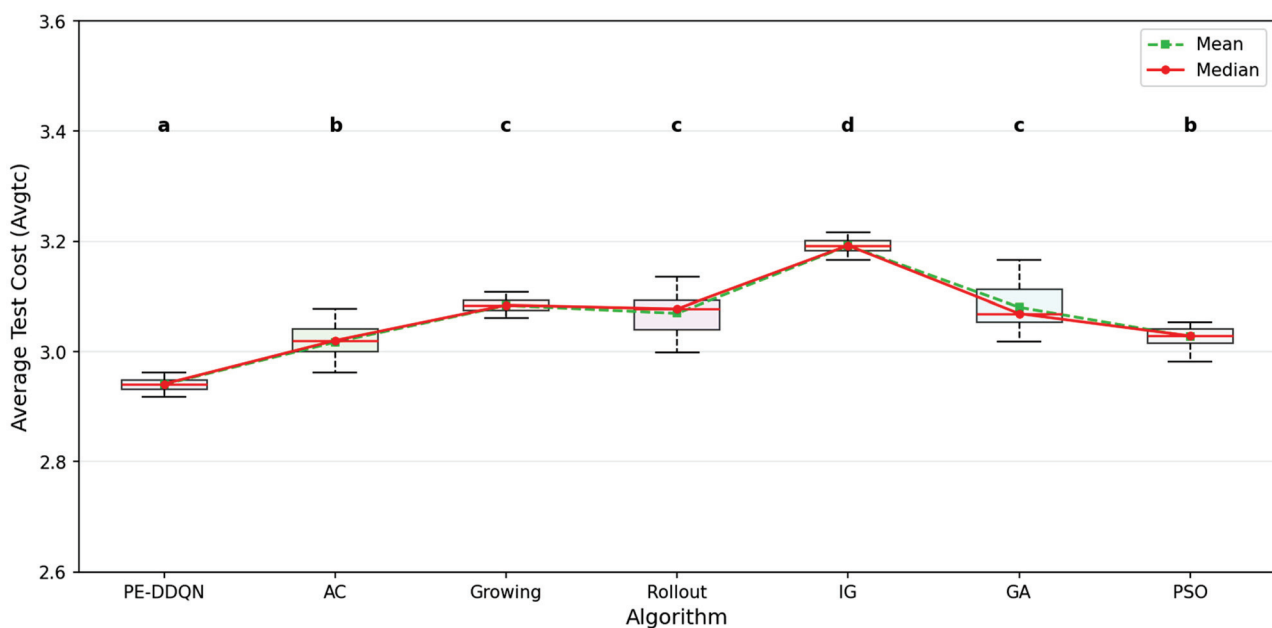


Figure 11. Comparison of test cost for 30 experiments (Dimension = 200).

We also conduct a series of the aforementioned experiments in other matrix dimensions. Among them, except for the PE-DDQN algorithm, the grouping situation of the other algorithms will change, indicating that they will fall into a local optimum as the matrix dimension increases, thereby leading to a decrease in algorithm stability. However, the PE-DDQN algorithm is always classified into group a, which is the group with the best performance, demonstrating the reliability of the proposed method.

The analysis results of the interaction effect show that the performance gap between algorithms is highly dependent on the dimensionality of the problem. To visually unravel this dynamic statistical divergence, Figure 12 illustrates the sequential evolution of the pairwise p -value against PE-DDQN across all evaluated dimensions. The vertical axis of

the figure shows pairwise comparison p -values on a logarithmic scale, and the red dashed line denotes the significance threshold of $p = 0.05$. The area above the red line corresponds to $p > 0.05$, indicating no statistically significant difference between algorithms, which is marked as “ns” (not significant) on the axis ticks. The single asterisk * at $p = 0.05$ represents a significant difference, the double asterisk ** at $p = 0.01$ represents a highly significant difference, and the triple asterisk *** at $p \leq 0.001$ represents an extremely significant difference. The red shaded region below the red line satisfies $p < 0.05$, demonstrating that PE-DDQN achieves statistically superior performance compared with other algorithms. As the dimension of the MVD matrix increases, the p -values corresponding to most comparison algorithms keep decreasing, which further strengthens the statistical significance of PE-DDQN’s superiority. At the initial stage, when the dimension is 20, advanced heuristics such as GA and PSO establish perfect statistical parity with PE-DDQN, confirming their adequacy for simple state spaces. However, as the combinatorial explosion of the multi-valued matrix intensifies, a collective downward trajectory into the significance domain ($p < 0.05$) is prominently observed. Heuristics like rollout and growing dive below the threshold early at very small dimensions, while GA and PSO manage to sustain equivalence up to 40 and 60 dimensions separately before undergoing a sharp drop. Eventually, at the very high complexity dimension, the p -value of all traditional algorithms has uniformly dropped to the lowest limit of 0.001. This dynamic significance evolution rigorously demonstrates that traditional methods suffer from acute performance decay under the curse of dimensionality, whereas PE-DDQN exhibits robust cost-suppression and outstanding architectural scalability for high-dimensional complex systems.

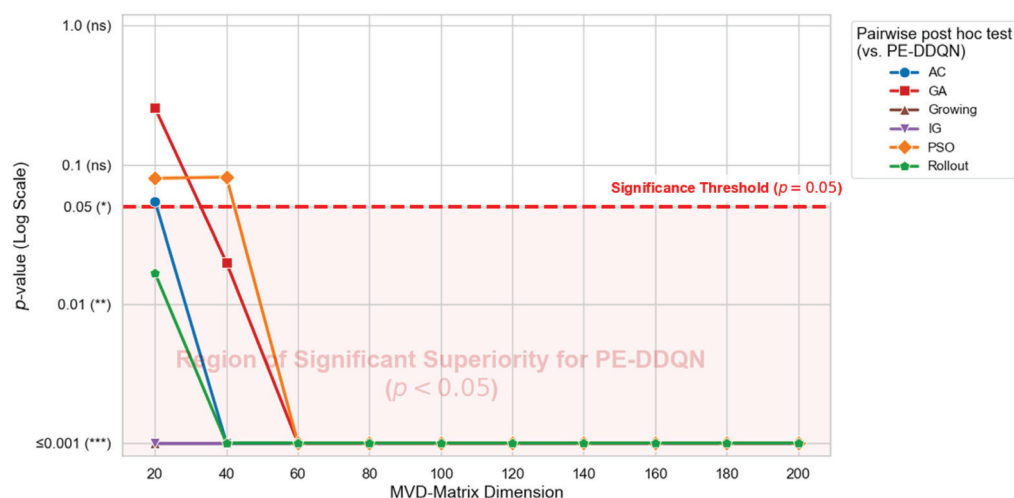


Figure 12. The changes in the p -values of each dimension.

Table 8 indicates that an increase in the dimension of the MVD matrix will lead to an increase in the test costs output by each algorithm. To compare the differences in the growth of test cost across various algorithms as dimensionality increases, we construct a multiple linear regression model incorporating interaction terms with PE-DDQN serving as the baseline. In this model, the interaction coefficient mathematically represents the exact difference in the fitted “dimension-cost” slopes between a traditional algorithm and PE-DDQN. We also calculate the ratio of each interaction coefficient to the fitted “dimension-cost” slope of the corresponding compared algorithm. This reduction ratio intuitively reflects the degree to which PE-DDQN outperforms traditional algorithms in mitigating the curse of dimensionality. These results are presented graphically using bar charts in Figure 13, and the significant p -values of the corresponding interaction terms are also annotated. This confirms that as the complexity of the system increases, the

growth rate of testing costs of the traditional algorithm is significantly faster than that of PE-DDQN. Among them, the PSO algorithm shows the fastest degradation rate as the dimension increases. The reduction ratio line also definitively shows that the average increase rate of the test cost output by PE-DDQN decreases by 12.6% to 31.6% compared to various traditional algorithms as the dimension of the MVD matrix increases, which proves that the PE-DDQN possesses a statistically superior capability to suppress cost growth in high-dimensional environments, outperforming every single baseline heuristic algorithm.

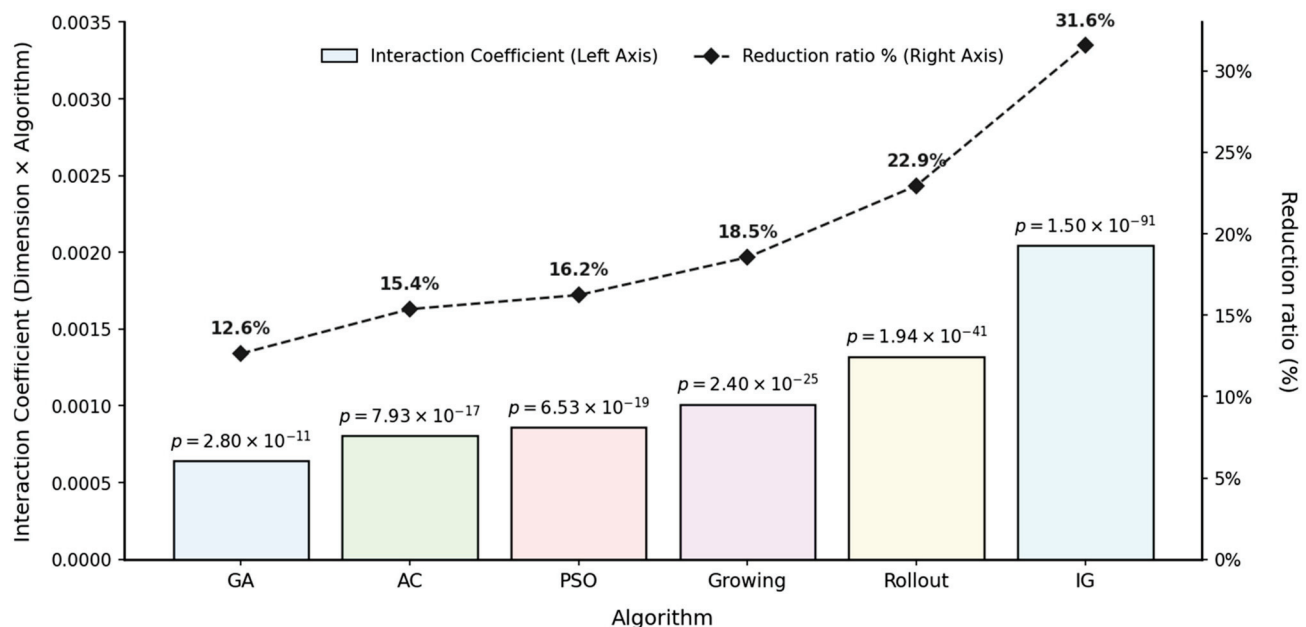


Figure 13. Interaction term coefficient and its significance compared with traditional algorithms.

To further evaluate the computational complexity and scalability of the proposed framework, Table 10 presents the average test time of different algorithms across varying MVD matrix dimensions from 20 to 200 (unit: seconds). All experiments in this study are conducted on a high-performance mobile workstation equipped with an Intel Core i7-11800H CPU @ 2.30 GHz (8 cores), 32 GB of DDR4 RAM (3200 MHz), and an NVIDIA GeForce RTX 3060 Laptop GPU. The proposed PE-DDQN framework and all comparative algorithms are implemented with CUDA enabled for hardware acceleration. As shown in Table 10, simple heuristic algorithms such as Growing and IG maintain extremely low computational times across all dimensions due to their greedy nature. However, as demonstrated in previous sections, this speed comes at the severe cost of falling into local optima and yielding poor diagnostic strategies. Conversely, complex heuristic algorithms like AC suffer from a severe combinatorial explosion, with test times surging exponentially to 457.13 as the dimension scales to 200.

It is noteworthy that population-based heuristics like GA and PSO maintain moderate computational times, which are lower than that of PE-DDQN. However, the optimization performance of these two common heuristic algorithms significantly deteriorates in high-dimensional scenarios, and their speed advantage is unable to compensate for the deficiency in their optimization performance.

Since PE-DDQN employs deep neural networks for fitting, its running time is longer compared to ordinary heuristic algorithms, but it remains within a controllable range overall. It effectively avoids the exponential time explosion seen in AC while dedicating a reasonable and controllable amount of computational time to successfully overcome the curse of dimensionality and find globally optimal strategies. Consequently, PE-DDQN

offers the best trade-off between computational scalability and optimization accuracy for complex systems.

Table 10. Comparison of average test time of different algorithms.

Dim	PE-DDQN	AC	Growing	Rollout	IG	GA	PSO
20	8.85	0.18	0.01	0.02	0.00	4.33	2.22
40	11.71	1.64	0.01	0.21	0.01	5.72	3.45
60	17.38	7.32	0.02	0.97	0.02	11.21	6.86
80	21.11	20.96	0.04	2.49	0.04	16.96	11.99
100	38.68	43.41	0.07	9.08	0.10	25.58	16.35
120	44.71	78.36	0.16	20.13	0.15	32.84	25.60
140	53.74	124.00	0.16	24.17	0.20	35.43	28.72
160	65.86	189.22	0.19	33.37	0.20	42.95	38.55
180	79.46	271.93	0.30	50.43	0.29	44.40	39.32
200	98.87	457.13	0.39	69.91	0.37	55.05	49.68

Overall, traditional heuristic optimization algorithms are prone to performance degradation and local optimum trapping when facing high-dimensional MVD matrix optimization problems. Benefiting from deep reinforcement learning-driven feature extraction and adaptive decision-making, the proposed PE-DDQN algorithm can effectively cope with the dimensionality curse. It achieves better test cost optimization performance while ensuring acceptable computational efficiency, which is more suitable for high-dimensional engineering application scenarios.

4.3.2. Ablation Experiment

In order to highlight the efficiency and stability of the convergence of PE-DDQN during the training process, DQN, DDQN, PE-DQN and PE-DDQN are respectively used to optimize the test points of a random high-dimensional MVD matrix, and the cost optimization objective is adopted. The average step-wise Q-values of the four algorithms during training are illustrated in Figure 14.

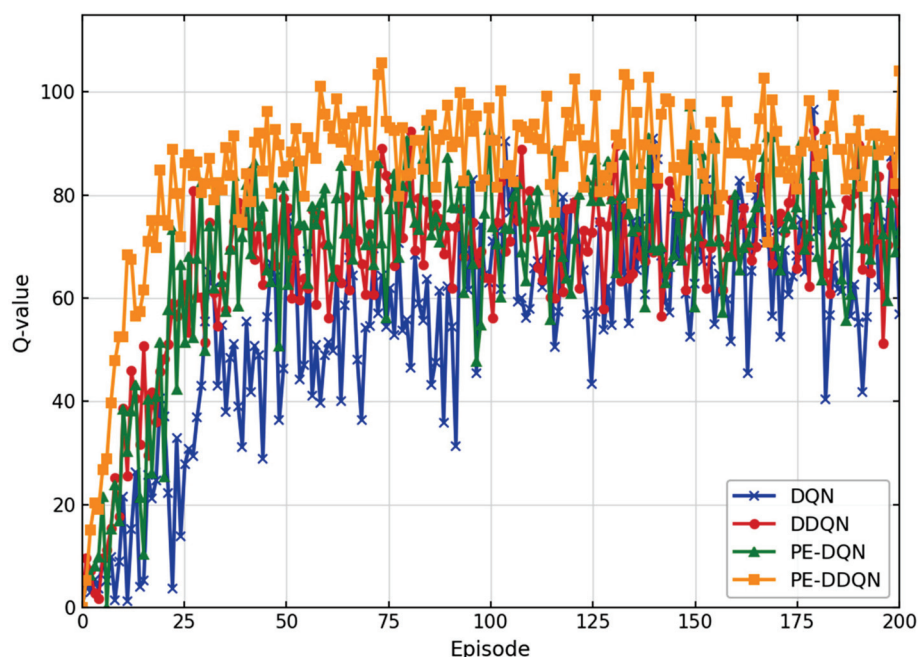


Figure 14. The average Q-value of the four algorithms.

As shown in Figure 14, DQN exhibits severe Q-value overestimation bias, which causes the agent to overprioritize actions with artificially inflated Q-values while neglecting potentially optimal action spaces. Consequently, it fails to explore high-reward action strategies, leading to low cumulative rewards, a globally suboptimal policy, and persistently low converged Q-values. By decoupling action selection from value evaluation, DDQN effectively alleviates overestimation and achieves a higher steady-state Q-value with reduced training fluctuations compared with DQN, yet its convergence speed remains relatively limited. Benefiting from the policy-enhanced module, PE-DQN achieves the fastest convergence rate by rapidly extracting critical optimization information from early training samples, though its training stability is inferior to DDQN.

Further integrating the advantages of policy enhancement and double-network structure, PE-DDQN not only inherits the rapid convergence characteristic of PE-DQN but also retains the low-fluctuation merit of DDQN. Its prioritized experience replay mechanism further accelerates the learning of high-impact key samples containing optimal action information. It can avoid wasting computational resources on low-value samples and guide more efficient policy optimization toward the global optimum. Overall, PE-DDQN achieves the best comprehensive performance in terms of convergence speed, training stability, and final Q-value level among all compared algorithms.

To verify the enhancement effect of the DDQN and PER module introduced in this paper on the testability strategy optimization, as well as the gain in adaptability of the model to a high-dimensional MVD matrix, we conduct test cost optimization experiments for the aforementioned ablation models. These experiments are under different MVD matrix dimensions with randomly generated matrix elements. For each matrix dimension ranging from 20 to 200, 30 independent repeated experiments are implemented for the four algorithms. Table 11 summarizes the average test cost and standard deviation.

Table 11. Comparison of average test time and standard deviation for 30 trials.

Dim	DQN	PE-DQN	DDQN	PE-DDQN
2.155 ± 0.0029	2.262 ± 0.0292	2.219 ± 0.0339	2.159 ± 0.0095	2.155 ± 0.0029
2.215 ± 0.0033	2.420 ± 0.0488	2.348 ± 0.0381	2.230 ± 0.0100	2.215 ± 0.0033
2.236 ± 0.0032	2.482 ± 0.0327	2.466 ± 0.0352	2.238 ± 0.0083	2.236 ± 0.0032
2.285 ± 0.0024	2.497 ± 0.0434	2.435 ± 0.0316	2.293 ± 0.0088	2.285 ± 0.0024
2.360 ± 0.0032	2.799 ± 0.1051	2.730 ± 0.0383	2.402 ± 0.0090	2.360 ± 0.0032
2.507 ± 0.0024	3.033 ± 0.0920	2.899 ± 0.1076	2.589 ± 0.0299	2.507 ± 0.0024
2.590 ± 0.0026	3.275 ± 0.0987	2.964 ± 0.1003	2.728 ± 0.0309	2.590 ± 0.0026
2.683 ± 0.0021	3.409 ± 0.1133	3.004 ± 0.0942	2.817 ± 0.0383	2.683 ± 0.0021
2.824 ± 0.0029	3.377 ± 0.1145	3.058 ± 0.0960	2.934 ± 0.0324	2.824 ± 0.0029
2.941 ± 0.0022	3.553 ± 0.1010	3.339 ± 0.1059	3.021 ± 0.0452	2.941 ± 0.0022

In terms of optimization performance, DQN and PE-DQN suffer from severe overestimation bias due to the absence of a double-network mechanism, which traps the models in suboptimal local minima and prevents them from obtaining globally optimal test strategies. Correspondingly, their larger standard error range in Figure 15 indicates poor stability and significant random fluctuations across repeated runs. By introducing the double-network structure, DDQN effectively mitigates overestimation and performs well in low-dimensional scenarios. Nevertheless, as the dimension increases, the diversity and redundancy of experience samples will grow considerably. Without prioritized sampling, DDQN exhibits degraded optimization performance and moderate stability in high-dimensional cases. By integrating the double-network structure and prioritized experience replay, PE-DDQN achieves the lowest average test cost across all dimensions,

accompanied by the narrowest standard error bars in Figure 15, which verifies its superior optimization accuracy and robustness.

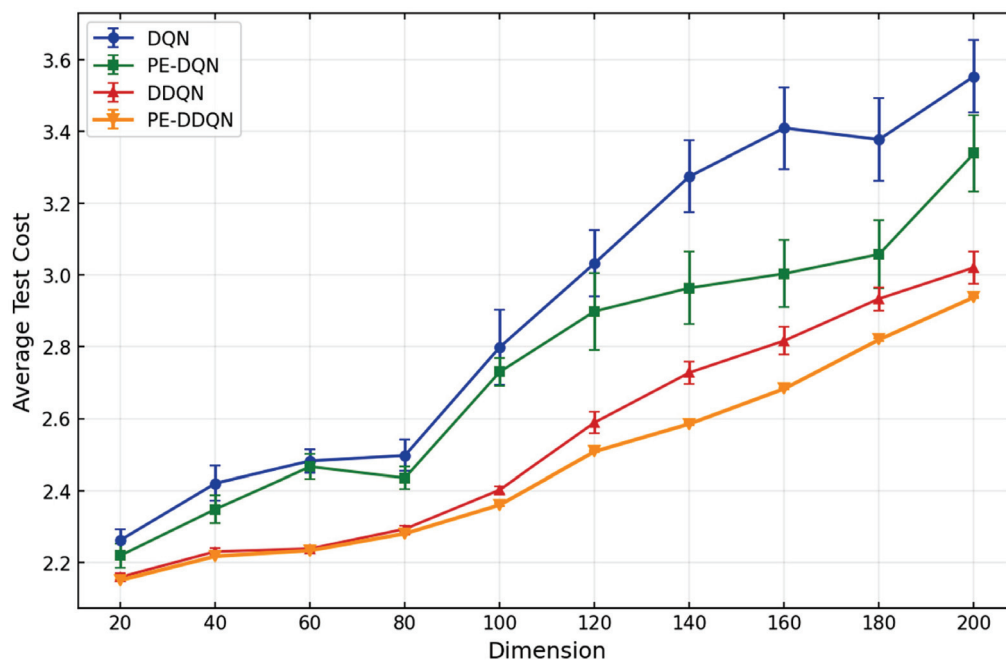


Figure 15. Comparison of average test cost of four algorithms under different dimensions.

To verify the advantages of PER and DDQN mechanisms in high-dimensional environments, a two-way ANOVA is performed on the test cost of each algorithm. The ANOVA analysis results for the main effects of the algorithms are $F = 140.8$ and $p < 0.001$. This indicates that there are significant differences among the algorithms as a whole. The analysis results of the interaction effect are $F = 7.060$ and $p < 0.001$, which reveals an extremely significant interaction effect between algorithm type and environmental dimension. Combined with Figure 15, it is the PE-DDQN model that outperforms the other three ablation models, and the advantage becomes more pronounced as the dimension increases.

Based on several ablation algorithms, we also construct a multivariate linear regression model with interaction terms. Taking the proposed PE-DDQN as the baseline, the interaction coefficients between it and each ablation model are calculated, as well as the reduction ratio of each interaction coefficient to the fitted “dimension-cost” slope of the corresponding ablation model, which is shown in Figure 16. The calculation results clearly demonstrate that the introduction of these two mechanisms indeed leads to a significant improvement in performance. DQN suffers the steepest performance degradation with growing dimensions without PER or DDQN constraints. The PE-DQN achieves a notably lower interaction coefficient by easing high-dimensional feature sparsity-induced learning bottlenecks via prioritized sampling. The DDQN separates action selection from value estimation. Its interaction coefficient is already relatively low, and there is no significant difference from PE-DDQN. However, Figure 15 shows that the introduction of PER to it can still reduce the test cost in a high-dimensional matrix environment. The reduction ratio line also definitively shows that the average increase rate of the test costs output by the PE-DDQN algorithm is approximately 42.7% lower than DQN when the dimension of the MVD matrix increases, and it is also lower than the other two ablation models.

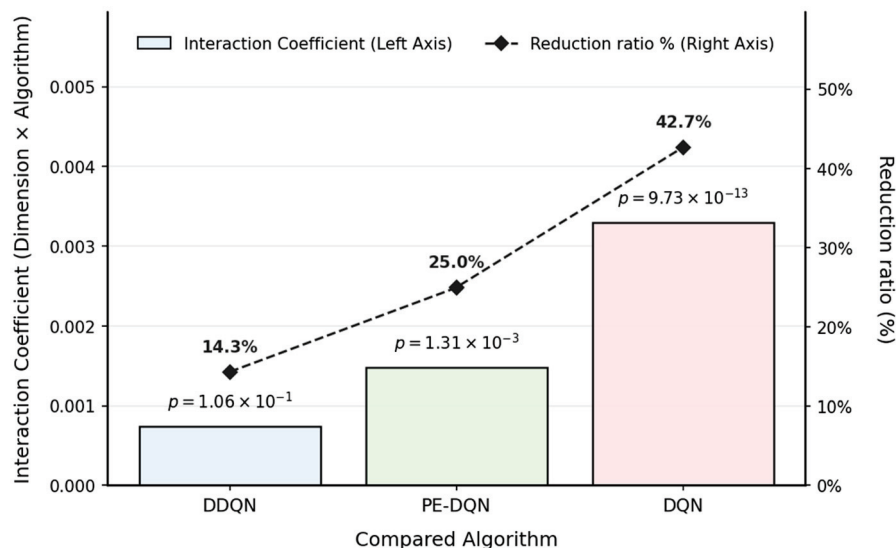


Figure 16. Interaction term coefficient and its significance compared with the ablation model.

From the above experiments, it can be seen that the introduction of PER and DDQN has a significant effect in high-dimensional environments. DQN typically struggles with severe overestimation bias and sample redundancy. The introduction of DDQN addresses the overestimation issue by decoupling action selection from value evaluation. This stabilization is crucial in large-scale testability environments, as it prevents the agent from overvaluing suboptimal test points and getting trapped in local minima during complex fault isolation sequences. On the other hand, the PER mechanism specifically tackles the combinatorial explosion of experience samples. By prioritizing the selection of high-value transfer states, PER ensures that the network does not waste computing resources on redundant or information-less test steps. Consequently, while DDQN provides a stable and accurate evaluation baseline, PER accelerates convergence and enhances the discovery of globally optimal strategies. Their synergistic integration into the PE-DDQN framework forms a highly persuasive and robust solution for large-scale testability strategy optimization.

In summary, the comprehensive experimental evaluations conducted in this section rigorously demonstrate the superiority of the proposed PE-DDQN framework from multiple perspectives. First, extensive comparative experiments are performed against six representative heuristic algorithms across a wide range of MVD matrix dimensions ranging from 20 to 200. Exhaustive statistical significance analyses, including ANOVA and pairwise post hoc tests, provide robust evidence that PE-DDQN achieves significantly lower testing costs ($p < 0.001$). Furthermore, multiple linear regression analyses incorporating interaction terms mathematically confirmed that PE-DDQN possesses a statistically superior capability to suppress the cost-escalating effect induced by the curse of dimensionality. The execution times of different algorithms also confirm that the proposed algorithm maintains an overall controllable computational complexity. It indicates that even in high-dimensional scenarios, the proposed PE-DDQN can derive a superior diagnostic strategy within an acceptable execution time. Finally, comprehensive ablation studies are conducted to explicitly isolate and quantify the individual contributions of the DDQN and PER mechanisms. The results rigorously proved their synergistic necessity in preventing overestimation bias and overcoming sample redundancy. Together, these thorough theoretical and empirical validations confirm that PE-DDQN offers a statistically superior, highly scalable, and structurally robust solution for complex system diagnostics.

5. Conclusions

This study proposed a novel testability strategy optimization method to address the issue of poor performance of the optimization strategy based on a high-dimensional MVD matrix. The proposed method is applied to the sampling circuit and the MIRFS, and a high-dimensional matrix simulation experiment is conducted to compare it with other algorithms. The main conclusions are as follows:

- (1) The Markov decision model in terms of test sequence length and test cost is effective. By linking testability strategy optimization with the Markov decision process, the optimization model is established. The sets of elements and reward functions for test sequence length and test cost optimization are established. The reward mechanisms enable the intelligent agent to effectively minimize the test sequence length or test cost while isolating all faults.
- (2) The construction of the DQN enables the agent to fit the Q-value of test points and select the optimal test point. A deep neural network is used to approximate the Q-value functions, and the selection of the testability strategy will be based on this neural network. Experience samples are obtained through the MVD matrix, and based on the Markov decision model, the parameters of the Q-network are updated to fit the Q-value through sample learning. The establishment of the DQN ensures that the model can select the optimal test point in each fault isolation status.
- (3) The introduced DDQN and PER mechanism can effectively enhance the model's adaptability to the high-dimensional matrix environment. The optimization algorithm based on PE-DDQN can effectively alleviate the overestimation problem of DQN and enable the model to focus on important samples, thereby improving the convergence. Thus, it significantly improves the training efficiency of the model and the reliability of the output testability strategy when dealing with a high-dimensional MVD matrix.

In summary, the overarching contribution of this study is the development of a comprehensive DRL-based framework structurally tailored for testability strategy optimization under multi-valued dependency conditions. By strategically adapting and integrating mature reinforcement learning mechanisms into the domain of testability modeling, this research successfully translates advanced artificial intelligence algorithms into practical engineering solutions. The proposed framework effectively addresses the curse of dimensionality and local optimum trapping inherent in traditional methods, providing a robust, highly efficient, and scalable diagnostic tool for modern complex systems.

The future work will focus on establishing an adaptive mechanism for adjusting the reward functions. The weights and parameters of the reward functions will be dynamically changed based on the real-time status of the system and historical performance. This enables the algorithm to better adapt to the dynamic changes in the system, thereby enhancing the effectiveness and adaptability of the reinforcement learning algorithm during its long-term operation.

Author Contributions: Conceptualization, C.Z. and F.W.; methodology, C.Z., Y.Z. and F.W.; software, Y.Z., X.S. and L.Z.; validation, Z.D. and L.Z.; formal analysis, X.S. and Z.D.; investigation, Y.Z.; data Curation, F.W. and Z.D.; writing—original draft, Y.Z.; writing—review and editing, C.Z., F.W., X.S. and L.Z.; supervision, C.Z. and X.S.; project administration, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the National Key Research and Development Program of China, grant number 2023YFF0719100; the National Key Scientific Research Projects of China, grant numbers EF6D4B9, JSZL2022607B002; and the Science and Technology Development Plan Project of Shaanxi Province, grant number 2024QCY-KXJ-038.

Data Availability Statement: The data that has been used is confidential.

Acknowledgments: The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

D-matrix	Dependency matrix
MVD matrix	Multi-valued dependency matrix
POMDP	Partially observable Markov decision process
DRL	Deep reinforcement learning
DQN	Deep Q-Network
DDQN	Double DQN
PER	Prioritized experience replay
PE-DDQN	Prioritized Experience Double Deep Q-Network
TD-error	Temporal difference error
MIRFS	Multi-functional integrated radio frequency system
RF	Radio frequency
IF	Intermediate frequency
PA	Power amplifier
AC	Ant colony
IG	Info-entropy greedy
GA	Genetic algorithm
PSO	Particle swarm optimization
ANOVA	Analysis of variance

References

1. Wang, W.; Gong, X.; Wang, X.; Cai, S.; Liu, P.; Xiong, N.N. DAF: An Effective Design-for-Testability Authorization Framework Based on Obfuscation Mechanisms for Defending Complex Attacks. *IEEE Internet Things J.* **2024**, *11*, 38829–38845. [CrossRef]
2. Ramirez, C.A.; Thompson, A.E.; Gorthala, R. A Design for Testability (DFT) Strategy for the Development of Highly Complex Safety-Critical System Using a Model-Based Systems Engineering (MBSE) Approach. In *Proceedings of the 2024 IEEE International Symposium on Systems Engineering (ISSE), Perugia, Italy, 16 October 2024*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8.
3. Zhang, C.; Cui, C.; Zhou, D.; Dong, Z.; He, S.; Zhou, Z. Phase-Based Testability Analysis of Multi-Function Integrated RF System Using Generalized Stochastic Petri Net. *Expert Syst. Appl.* **2025**, *284*, 127819. [CrossRef]
4. Wang, P.; Yu, Y.; Li, X. Testability Modeling and Test Point Optimization Method of Multi-State System. *IEEE Access* **2020**, *8*, 155011–155019. [CrossRef]
5. Deb, S.; Pattipati, K.R.; Raghavan, V.; Shakeri, M.; Shrestha, R. Multi-Signal Flow Graphs: A Novel Approach for System Testability Analysis and Fault Diagnosis. In *Proceedings of the AUTOTESTCON '94, Anaheim, CA, USA, 20–22 September 1994*; IEEE: Piscataway, NJ, USA, 1994; pp. 361–373.
6. Yang, C.; Zhong, A.; Yang, X.; Chen, Y. Graphical Hybrid Diagnostic Model-Based Software Architecture for System Testability Analysis. In *Proceedings of the 2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway, 5–8 August 2024*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
7. Lv, D.; Wu, H.; Qiu, G.; Cai, Y. Testability Modeling and Analysis of Layered System Based on Multi Signal Flow Graph. In *Proceedings of the 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 14–16 April 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 1295–1299.
8. Li, X.; Cui, W.; Li, L.; Yang, J. A Review on Dependency Matrix and Its Application in Fault Diagnosis. In *Proceedings of the 2022 Global Reliability and Prognostics and Health Management (PHM-Yantai), Yantai, China, 13–16 October 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8.
9. Zhang, J.; Chen, D.; Gao, P.; Zhang, J.; Zhang, Y. A Dependency Matrix Processing Algorithm to Prioritize High Incidence Faults. *J. Syst. Eng. Electron.* **2025**, *36*, 962–971. [CrossRef]
10. Luo, H.; Wang, Y.; Lin, H.; Jiang, Y. A New Optimal Test Node Selection Method for Analog Circuit. *J. Electron. Test.* **2012**, *28*, 279–290. [CrossRef]

11. Liu, W.M.; Chen, C.; Lin, Y.G.; Ye, X.R.; Zhai, G.F.; Lin, R.S.; Zheng, W. Testability Design of Power Distribution System Considering Noise Immunity Characteristics. *Microelectron. Reliab.* **2023**, *150*, 115178. [CrossRef]
12. Tian, H.; Duan, F.; Fan, L.; Sang, Y. Fault Diagnostic Strategy of Multivalued Attribute System Based on Growing Algorithm. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2019**, *233*, 235–245. [CrossRef]
13. Liu, Y.; Wu, Z.; Deng, J.; Zhou, C. Processing of Multi-Valued Attributes Based on Sparse Matrix. In *Proceedings of the 2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS), Nara, Japan, 24–26 November 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 318–320.
14. Huang, S.; Cai, J.; Dai, D. Testability Modeling of Aeroengine and Analysis Optimization Method Based on Improved Correlation Matrix. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2025**, *239*, 46–58. [CrossRef]
15. Su, Y.; Liang, X.; Gu, C.; Khemani, V.; Pecht, M. A Multivalued Test and Diagnostic Strategy Optimization Method for Aircraft System Fault Diagnosis. In *Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
16. Zhu, X.; Sun, J.; Yan, Z.; Xu, Y. Integrated Testability Modeling Method of Complex Systems for Fault Feature Selection and Diagnosis Strategy Optimization. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2025**, *239*, 443–458. [CrossRef]
17. Yang, C.; Tian, S.; Long, B. Application of Heuristic Graph Search to Test-Point Selection for Analog Fault Dictionary Techniques. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 2145–2158. [CrossRef]
18. Mei, W.; Liu, Z.; Tang, L.; Su, Y. Test Strategy Optimization Based on Soft Sensing and Ensemble Belief Measurement. *Sensors* **2022**, *22*, 2138. [CrossRef] [PubMed]
19. Han, L.; Shi, X.; Wang, T. Bayesian Network Model Test Configuration Method Based on Genetic and Binary Discrete Particle Swarm Combination Algorithm. *J. Phys. Conf. Ser.* **2020**, *1642*, 012007. [CrossRef]
20. Yang, C.; Su, R.; Long, B. Methods of Sequential Test Optimization in Dynamic Environment. *Microelectron. Reliab.* **2017**, *70*, 112–121. [CrossRef]
21. Wang, J.; Liu, Z.; Wang, J.; Long, B.; Zhou, X. A General Enhancement Method for Test Strategy Generation for the Sequential Fault Diagnosis of Complex Systems. *Reliab. Eng. Syst. Saf.* **2022**, *228*, 108754. [CrossRef]
22. Liang, Y.; Xiao, M.; Wang, X.; Tang, X.; Zhu, H.; Li, J. A POMDP-Based Optimization Method for Sequential Diagnostic Strategy with Unreliable Tests. *IEEE Access* **2019**, *7*, 75389–75397. [CrossRef]
23. Cui, W.; Li, L.; Li, H.; Guan, F.; Hong, D.; Liu, G. Dependency Matrix-Based Fast Fault Diagnosis Using Search Algorithm. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3510908. [CrossRef]
24. Zhang, J.; Chen, D.; Gao, P. A Divide-and-Conquer Information Entropy Algorithm for Dependency Matrix Processing. *IEEE Access* **2023**, *11*, 121306–121313. [CrossRef]
25. Wang, J.; Liu, Z.; Yao, X.; Wang, Y.; Li, Q.; Mi, J. A Sequential Diagnostic Strategy Generation Transformation Method for Large-Scale Systems Based on Multi-Signal Flow Graph Model and Multi-Objective Optimization. *Reliab. Eng. Syst. Saf.* **2025**, *259*, 110922. [CrossRef]
26. Dandan, H.; Ningyun, L.; Bin, J. A Test Sequencing Optimization Method for Fault Diagnosis of Engineering Systems. In *Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, 28–30 May 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 1208–1213.
27. Tian, H.; Xu, R.B.; Jiang, Y.H.; Zhang, W.H.; Deng, S.E. Fault Diagnosis Strategy of Multi-Valued Attribute System Based on Discrete Particle Swarm Optimization Algorithm. *Acta Armamentarii* **2022**, *43*, 3240. [CrossRef]
28. Saied, M.; Tahan, N.; Chreif, K.; Francis, C.; Noun, Z. Development of Deep Reinforcement Learning-Based Fault Diagnosis Method for Actuator Faults in Unmanned Aerial Vehicles. *Aeronaut. J.* **2025**, *129*, 1303–1319. [CrossRef]
29. Liu, Z.; Ma, L.; Wang, K.; Zhang, J.; Si, C.; Yi, J.; Mu, C. Uncertainty-Aware Model-Based Multi-Agent Deep Reinforcement Learning for Robust Active Voltage Control. *IEEE Trans. Circuits Syst. I* **2026**, *73*, 618–629. [CrossRef]
30. Wang, X.; Wang, S.; Liang, X.; Zhao, D.; Huang, J.; Xu, X.; Dai, B.; Miao, Q. Deep Reinforcement Learning: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 5064–5078. [CrossRef] [PubMed]
31. Cao, C.; Zhang, Q.; Li, M.; Wang, S. Flame Combustion State Recognition in Municipal Solid Waste Incineration Processes Based on Image Multi-Threshold Segmentation and DQN-PL Model. *Energy* **2025**, *331*, 136967. [CrossRef]
32. Zhang, J.; Chen, H.; Sun, H.; Xu, H.; Yan, T. Convolutional Neural Network-Based Deep Q-Network (CNN-DQN) Path Planning Method for Mobile Robots. *Intell. Serv. Robot.* **2025**, *18*, 929–950. [CrossRef]
33. Li, C.; Wu, Z. F-DQN: An Optimized DQN for Decision-Making of Generator Start-up Sequence after Blackout. *Appl. Intell.* **2024**, *54*, 5521–5535. [CrossRef]
34. M. Ali Al-Shamiri, M.; Ali, G.; Zain Ul Abidin, M.; Adeel, A. Multi-Attribute Group Decision-Making Method under Spherical Fuzzy Bipolar Soft Expert Framework with Its Application. *Comput. Model. Eng. Sci.* **2023**, *137*, 1891–1936. [CrossRef]
35. Wang, J.; Wang, J.; Zhang, X.; Liu, J. Feature Selection Based on Information Entropy with Variable Precision Fuzzy Mixed Granularity. *Appl. Intell.* **2025**, *55*, 999. [CrossRef]

36. Wang, J.; Li, X.; Wang, P.; Jin, X.; Yao, S.; Miao, S. CS-DRL: A Soft Policy Update Approach for Wireless Bandwidth Allocation Using Deep Reinforcement Learning. *Future Gener. Comput. Syst.* **2026**, *177*, 108215. [CrossRef]
37. Hamid, O.H. A Model-Based Markovian Context-Dependent Reinforcement Learning Approach for Neurobiologically Plausible Transfer of Experience. *HIS* **2015**, *12*, 119–129. [CrossRef]
38. Li, Y.; Zhang, Y.; Ding, Y.; Jiang, S.; Yuan, G. A Class Integration Test Order Generation Approach Based on Sarsa Algorithm. *Autom. Softw. Eng.* **2024**, *31*, 7. [CrossRef]
39. Garí, Y.; Pacini, E.; Robino, L.; Mateos, C.; Monge, D.A. Online RL-Based Cloud Autoscaling for Scientific Workflows: Evaluation of Q-Learning and SARSA. *Future Gener. Comput. Syst.* **2024**, *157*, 573–586. [CrossRef]
40. Kim, J.; Mo, J. A Dual-Critic DQN Architecture for Lifter Assignment in Multi-Floor Semiconductor FAB. *Int. J. Prod. Res.* **2025**, *63*, 1450–1472. [CrossRef]
41. Mohi Ud Din, N.; Assad, A.; Ul Sabha, S.; Rasool, M. Optimizing Deep Reinforcement Learning in Data-Scarce Domains: A Cross-Domain Evaluation of Double DQN and Dueling DQN. *Int. J. Syst. Assur. Eng. Manag.* **2024**. [CrossRef]
42. Hamid, O.H.; Braun, J. Reinforcement Learning and Attractor Neural Network Models of Associative Learning. In *Computational Intelligence*; Sabourin, C., Merelo, J.J., Madani, K., Warwick, K., Eds.; Studies in Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2019; Volume 829, pp. 327–349, ISBN 978-3-030-16468-3.

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.

Article

DDEC: Dual Dependency-Enhanced Contrastive Learning for Sparse Hypergraph Node Classification

Meilin Liu ¹, Wenping Zheng ^{1,2,*} and Shuxia Yuan ¹

¹ School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China; 202112407009@email.sxu.edu.cn (M.L.)

² Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, China

* Correspondence: wpzheng@sxu.edu.cn

Abstract

Hypergraph neural networks have shown strong potential for node classification due to their ability to capture high-order relationships and multi-granularity structural patterns. However, real-world hypergraphs are often sparse, which limits interaction modeling through node–hyperedge incidence and, in turn, weakens reliable attribute propagation and global dependency capture. To address this issue, we propose DDEC, a Dual Dependency-Enhanced Contrastive learning framework for sparse hypergraph node classification. To compensate for relational information lost under sparse structures, DDEC introduces an attribute view to complement the structural view. Since attribute information can be noisy and unreliable, we first design an entropy-guided feature recalibration mechanism to estimate node uncertainty and emphasize trustworthy attribute interactions. Building upon this, DDEC performs dual dependency enhancement from both structural and attribute perspectives. Specifically, we exploit the duality between a hypergraph and its line graph to perform line-graph transformation in both views, thereby constructing a shared dual relational space for interaction enhancement under sparse topologies. Within this dual space, we perform attention-based dependency enhancement in both views, so that the structural view captures explicit topological dependencies among hyperedges, while the attribute view uncovers latent semantic correlations beyond sparse incidence relations. The resulting representations from the two views are then adaptively fused, and collaborative contrastive learning is further performed at both the node and hyperedge levels to enforce multi-granularity semantic consistency. Experiments on eight public datasets demonstrate that DDEC consistently outperforms competitive baselines, validating its effectiveness and robustness.

Keywords: hypergraph neural networks; sparse hypergraph; node classification; dependency enhancement; contrastive learning

1. Introduction

With the rapid proliferation of complex networked data, modeling high-order relationships among entities has become increasingly important in representation learning. Although conventional graphs have achieved notable success in a wide range of machine learning tasks, they are inherently limited to pairwise interactions and therefore often fall short in representing real-world systems involving multi-entity relations. Hypergraphs provide a natural and expressive alternative by allowing each hyperedge to connect an

arbitrary number of nodes [1,2]. Owing to this advantage, hypergraph neural networks (HGNNs) have demonstrated strong effectiveness in various tasks, such as node classification, clustering, and recommendation [3–5].

Most existing HGNNs learn node representations through message passing over node–hyperedge incidence structures, where nodes aggregate high-order contextual information from their incident hyperedges. While this paradigm has achieved promising results, its effectiveness can deteriorate substantially when the underlying hypergraph is sparse. In many real-world scenarios, hypergraphs exhibit sparse connectivity [6], manifested as limited node–hyperedge interactions, weak hyperedge overlap, and fragmented structural organization. Such sparsity weakens the propagation of high-order information, reduces the reliability of local structural evidence, and hinders the modeling of long-range or implicit dependencies among hyperedges. As a consequence, node representations are often dominated by incomplete local observations, while unreliable attribute signals or structure-induced uncertainty may be propagated and amplified during message passing.

Despite the growing body of research on hypergraph learning, most existing methods still rely predominantly on local aggregation over the node–hyperedge bipartite structure [7]. This design becomes particularly vulnerable under sparse hypergraph settings for three reasons. First, sparse incidence patterns provide insufficient relational evidence for capturing dependencies beyond local neighborhoods, making it difficult to model latent interactions among hyperedges from structure alone. Second, most methods do not explicitly assess the reliability of attribute information. When raw features are weakly discriminative or incomplete, directly using them to construct auxiliary relations may introduce unreliable semantic neighbors, especially when structural support is weak. Third, semantic learning at different granularities is often handled in a decoupled manner, with limited coordination between node-level local consistency and hyperedge-level global semantics. These limitations call for a unified framework that can complement sparse structural evidence, strengthen trustworthy dependency modeling, and preserve semantic consistency across multiple granularities.

To this end, we propose DDEC, a Dual Dependency-Enhanced Contrastive learning framework for sparse hypergraph node classification. The central idea of DDEC is to improve representation learning by jointly exploiting structural and attribute information. When sparse incidence relations provide insufficient structural cues, the attribute view offers complementary semantic evidence for interaction modeling. Since attribute information may itself be noisy and unreliable, we first develop an entropy-guided feature recalibration mechanism to estimate node uncertainty and emphasize trustworthy attribute interactions. Building on this, we exploit the duality between a hypergraph and its line graph to perform line-graph transformation in both views, thereby constructing a shared dual relational space for interaction enhancement under sparse topologies. Within this dual space, attention-based dependency enhancement is performed in both views, so that the structural view captures explicit topological dependencies among hyperedges, while the attribute view uncovers latent semantic correlations beyond sparse incidence relations. The resulting representations from the two views are then adaptively fused. Furthermore, we design a collaborative contrastive learning strategy at both the node and hyperedge levels to enforce multi-granularity semantic consistency, thereby improving the robustness and discriminability of the learned representations.

Extensive experiments on eight public datasets show that DDEC consistently outperforms competitive baselines. Additional ablation and robustness analyses further validate the effectiveness and stability of the proposed framework.

The main contributions of this work are summarized as follows:

1. We propose DDEC, a unified framework for sparse hypergraph node classification that improves representation learning by jointly exploiting structural and attribute information.
2. We design an entropy-guided feature recalibration mechanism together with a dual-space dependency enhancement scheme to strengthen trustworthy interactions and improve dependency modeling under sparse hypergraph structures.
3. We develop a collaborative contrastive learning strategy at both node and hyperedge levels to enforce multi-granularity semantic consistency and enhance the robustness and discriminability of learned representations.
4. Extensive experiments on eight benchmark datasets, along with ablation and robustness studies, demonstrate the superiority, effectiveness, and stability of DDEC.

2. Related Work

Existing studies relevant to our work can be broadly grouped into two research lines: hypergraph neural networks and contrastive hypergraph learning. The former mainly focuses on high-order information propagation and dependency modeling over hypergraph structures, while the latter further introduces contrastive self-supervision to improve representation quality and semantic consistency.

2.1. Hypergraph Neural Networks

Hypergraph neural networks extend conventional graph learning beyond pairwise relations and provide a natural framework for modeling high-order interactions. Early studies mainly generalize graph convolution to node–hyperedge incidence structures. Representative models such as HGNN [7] establish the basic hypergraph propagation paradigm, while subsequent methods [8–11] improve it from different perspectives. UniGNN [12] further provides a unified formulation for graph and hypergraph neural networks. Although these methods improve hypergraph representation learning in various ways, they still largely rely on the original incidence structure and local propagation patterns.

To move beyond fixed propagation schemes, later studies explore more flexible structure modeling and adaptive hypergraph learning. For example, DHGNN [13] improves adaptability through dynamic hypergraph construction, DHKH [14] jointly models high-order and key structures, TDHNN [15] and HERALD [16] enhance representation quality through dynamic hyperedge modeling and Laplacian optimization, respectively, and MHSL [17] captures long-range dependencies via dynamic hypergraph construction. These methods enhance structural flexibility and better adapt hypergraph structures to downstream tasks. However, they still mainly refine node representations within the original or dynamically updated structure, with limited attention to explicit inter-hyperedge dependency modeling and semantic coordination across different granularities.

More recent studies seek to mitigate the challenges posed by sparse hypergraphs through auxiliary structural priors or explicit inter-hyperedge modeling. DPHGNN [18] incorporates low-order graph relations to alleviate feature degradation, and THTN [19] enhances cross-region interaction by introducing global nodes. Meanwhile, several methods explicitly model hyperedge relations beyond node-level aggregation. HeIHNN [20] constructs a hyperedge interaction graph, LHCN [21] performs hyperedge-level propagation through line-graph transformation, and HyperGT [22] and SHT [23] capture more complex hyperedge dependencies through global attention and learnable relation matrices, respectively. These studies highlight the importance of inter-hyperedge relation modeling when local node–hyperedge propagation becomes insufficient.

Nevertheless, existing methods still face several limitations in sparse settings. First, many approaches infer inter-hyperedge relations mainly through shared nodes or prede-

defined structural rules, which restricts their ability to capture latent semantic dependencies beyond explicit structural overlap. Second, they generally do not provide a unified mechanism for jointly modeling topological and semantic dependencies when global structural evidence is fragmented. Third, most methods do not explicitly assess the reliability of propagated node features, making them vulnerable to noisy or uncertain signals when structural evidence is sparse. In contrast, DDEC combines uncertainty-aware feature recalibration with dual-space dependency enhancement, enabling it to suppress unreliable node signals while complementing incomplete inter-hyperedge dependencies in sparse hypergraphs.

2.2. Contrastive Hypergraph Learning

Contrastive learning has become an effective self-supervised paradigm for representation learning by enforcing consistency across different views and improving representation discriminability [24]. In hypergraph learning, it provides complementary supervision beyond labeled data and helps mitigate representation degradation caused by insufficient supervision and sparse structural connectivity.

Early contrastive hypergraph methods mainly focus on constructing multiple views for self-supervised optimization. For example, CCL [25] jointly exploits graph and hypergraph views, HyperGCL [26] combines perturbation-based and generative views, HSL [27] preserves consistency between optimized and original hypergraphs through intra-hyperedge contrast, and HCN [28] improves unsupervised hypergraph clustering via multi-view generation and structural alignment. These studies show that contrastive learning can provide useful supervisory signals beyond labels. However, many of them still emphasize node-level alignment, with limited explicit modeling of semantic consistency at the hyperedge level.

A subsequent line of work extends contrastive learning from node-level alignment to joint node–hyperedge modeling. TriCL [29], CHGNN [30], HyGCL-AdT [31], and HSCL [32] enhance high-order semantic modeling through node masking, hyperedge perturbation, multi-view augmentation, and hierarchical contrastive objectives. Although these methods provide richer self-supervised signals, they still rely substantially on augmentation-based view construction. Under sparse hypergraph settings, such perturbations may increase view diversity, but they may also further weaken high-order structural relations.

Another line of work further strengthens contrastive hypergraph learning by introducing structure-aware inductive biases. Specifically, H2SeqRec [33] models multi-scale temporal hypergraphs in hyperbolic space, OFSH [34] preserves core high-order semantics through key-node-guided adaptive augmentation, DHCN [35] improves sparse interaction modeling via line-graph transformation, and MCHNN [36] enhances representation learning under sparse association settings through multiple negative-sample hypergraphs. These methods demonstrate the benefit of integrating structural modeling with contrastive objectives, but they still address structural incompleteness and semantic consistency in a relatively fragmented manner.

Overall, existing contrastive hypergraph learning methods remain limited by insufficient multi-granularity contrastive modeling, perturbation-based view construction that may damage sparse structural relations, and inadequate exploitation of latent inter-hyperedge dependencies. In contrast, DDEC integrates uncertainty-aware feature recalibration, dual-space dependency enhancement, and collaborative contrastive learning at both the node and hyperedge levels within a unified framework. As a result, it improves robustness to noisy signals, complements fragmented inter-hyperedge dependencies, and promotes semantic consistency across local and global structural granularities.

3. Preliminaries

3.1. Hypergraph Definition and Problem Formulation

A hypergraph is a high-order generalization of an ordinary graph and can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ denotes the set of n nodes and $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ denotes the set of m hyperedges. Different from a conventional graph in which each edge connects only two nodes, each hyperedge in a hypergraph can simultaneously connect an arbitrary number of nodes, thereby naturally modeling high-order relations. The topology of a hypergraph is commonly described by an incidence matrix $\mathbf{H} \in \mathbb{R}^{n \times m}$, where

$$H_{ij} = \begin{cases} 1, & \text{if } v_i \in e_j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Based on $\mathbf{H}$, the degree of node v_i and the cardinality of hyperedge e_j are defined as $d(v_i) = \sum_{e_j \in \mathcal{E}} H_{ij}$, $\delta(e_j) = \sum_{v_i \in \mathcal{V}} H_{ij}$. Accordingly, the corresponding diagonal node-degree matrix and hyperedge-degree matrix are denoted by $\mathbf{D}_v$ and $\mathbf{D}_e$, respectively. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote the initial node feature matrix, where d is the feature dimension.

To describe the sparsity characteristics of a hypergraph from the incidence perspective, we further consider two descriptive statistics, namely the average node participation and the average hyperedge cardinality:

$$\rho_v = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} d(v_i), \quad \rho_e = \frac{1}{|\mathcal{E}|} \sum_{e_j \in \mathcal{E}} \delta(e_j). \quad (2)$$

where ρ_v measures the average number of hyperedges incident to each node, while ρ_e measures the average number of nodes contained in each hyperedge. Although these two quantities are coupled through the total number of node–hyperedge incidences, they reflect sparsity from the node and hyperedge perspectives, respectively. Small values of ρ_v and ρ_e indicate limited node participation and restricted hyperedge scope, which together weaken high-order information propagation and make dependency modeling more challenging.

Given a partially labeled hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node feature matrix $\mathbf{X}$ and a labeled node set $\mathcal{V}_L \subset \mathcal{V}$, the task of hypergraph node classification is to learn a prediction function $f(\mathcal{G}, \mathbf{X})$ that infers the labels of unlabeled nodes in $\mathcal{V} \setminus \mathcal{V}_L$. In sparse hypergraphs, the key challenge lies in learning discriminative node representations under insufficient structural interactions and unreliable dependency propagation.

3.2. Dual Line-Graph Transformation

Given a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, its dual line graph is defined as $\mathcal{L}(\mathcal{G}) = (\mathcal{V}_L, \mathcal{E}_L)$, where each node in $\mathcal{V}_L$ corresponds to a hyperedge in $\mathcal{E}$, i.e., $\mathcal{V}_L = \mathcal{E}$. Two dual nodes are connected if their corresponding hyperedges overlap. Formally, the edge set is defined as $\mathcal{E}_L = \{(e_i, e_j) \mid e_i, e_j \in \mathcal{E}, e_i \cap e_j \neq \emptyset, i \neq j\}$. Thus, hyperedge overlap in the original hypergraph is converted into explicit adjacency in the dual line graph. As illustrated in Figure 1, this transformation maps each hyperedge to a node and represents overlap between hyperedges as graph connectivity.

To quantify the correlation strength between hyperedges, we adopt the Jaccard coefficient as the edge weight in the dual line graph:

$$w_{ij} = \frac{|e_i \cap e_j|}{|e_i \cup e_j|}, \quad (3)$$

where w_{ij} measures the degree of overlap in node membership between hyperedges e_i and e_j , and a larger value of w_{ij} indicates a stronger topological association between them.

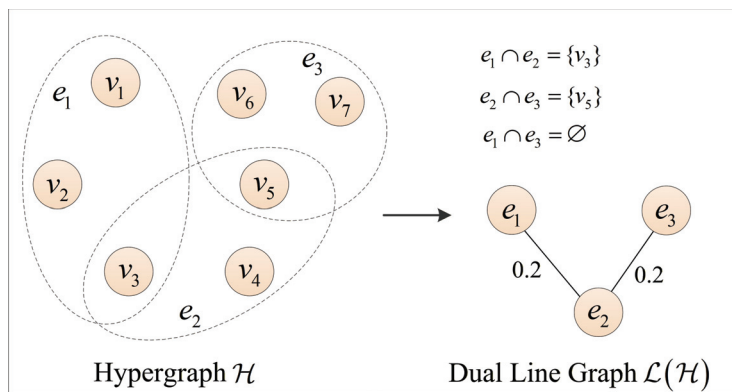


Figure 1. Illustration of a hypergraph and its dual line-graph transformation.

3.3. Hypergraph Convolution Propagation

Given a hypergraph $\mathcal{G}$, a standard one-layer hypergraph convolution can be formulated as

$$\mathbf{X}^{(l+1)} = \sigma \left(\mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W}_e \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-\frac{1}{2}} \mathbf{X}^{(l)} \boldsymbol{\Theta}^{(l)} \right), \quad (4)$$

where $\boldsymbol{\Theta}^{(l)}$ is the learnable parameter matrix at layer l , $\mathbf{W}_e$ is the hyperedge weight matrix, usually initialized as an identity matrix, $\sigma(\cdot)$ is a nonlinear activation function, and $\mathbf{X}^{(l+1)}$ denotes the output node representations. According to Equation (4), the propagation first aggregates node features into the hyperedge space through $\mathbf{H}^\top$, performs information transformation at the hyperedge level, and then projects the updated hyperedge information back to the node space through $\mathbf{H}$. In this way, a complete node–hyperedge–node message passing process is achieved, which provides the basic foundation for hypergraph representation learning.

4. Method

To address the challenges of sparse hypergraph node classification, we propose Dual Dependency-Enhanced Contrastive Learning (DDEC). As illustrated in Figure 2, DDEC consists of three key components: entropy-guided feature recalibration, dual dependency enhancement from the attribute and structural views, and multi-granularity collaborative contrastive learning. Specifically, DDEC first performs entropy-guided feature recalibration to suppress unreliable attribute signals under sparse structures. It then conducts dual dependency enhancement from both the attribute and structural views to complement missing relational information and model inter-hyperedge dependencies. Finally, a multi-granularity collaborative contrastive learning strategy is introduced to preserve semantic consistency and improve representation discriminability at both the node and hyperedge levels.

4.1. Entropy-Guided Feature Recalibration

In sparse hypergraphs, limited node–hyperedge interactions and weak hyperedge overlap make structural evidence insufficient for reliable representation learning. To compensate for missing relational information, we introduce an attribute view as a complementary source. However, raw node features may contain ambiguous, weakly discriminative, incomplete, or heterogeneous signals that cannot be fully resolved by the sparse structural context. Such unreliable signals may be amplified during message passing and impair subsequent attribute-based dependency modeling. This implicit noise is dataset-intrinsic rather than artificially injected and reflects the uncertainty in how much each node can reliably contribute to subsequent attribute interactions. To mitigate this effect, we perform

entropy-guided feature recalibration before constructing the attribute view, suppressing unreliable signals and providing a more robust basis for dual dependency enhancement.

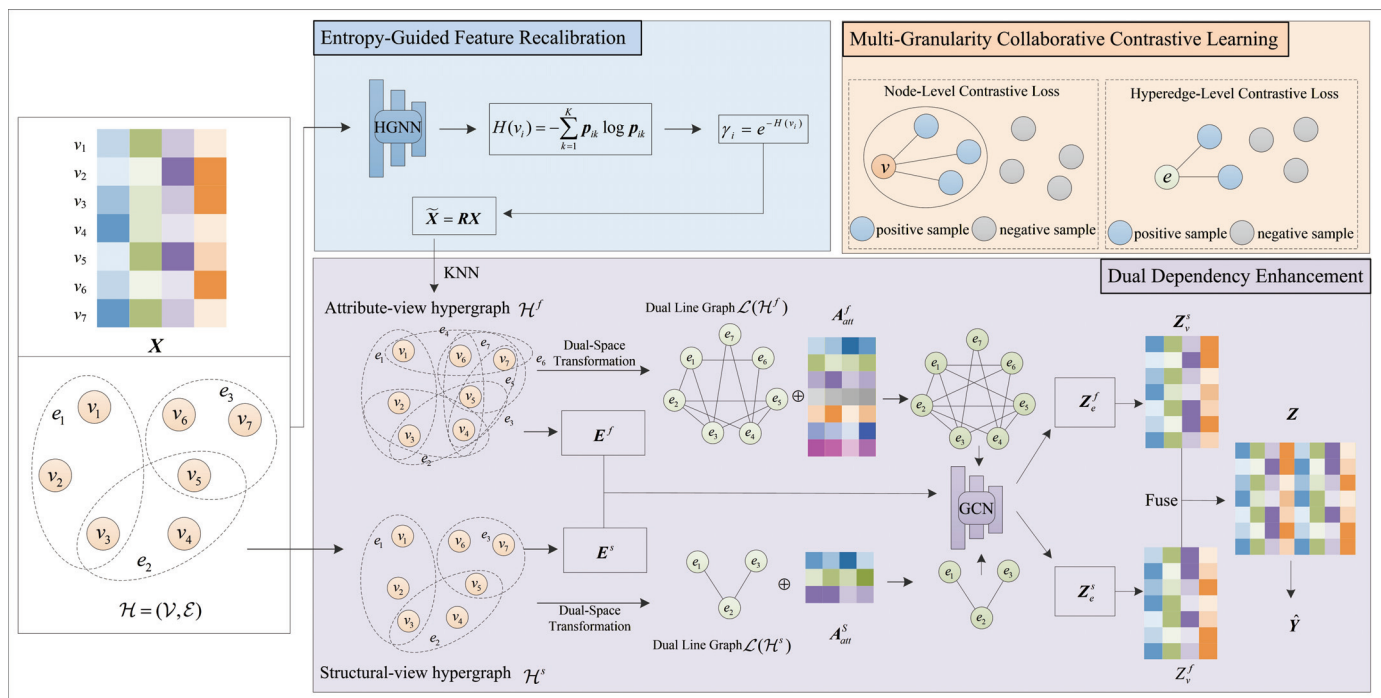


Figure 2. Framework of DDEC.

To estimate node-level feature reliability, we employ a lightweight auxiliary hypergraph predictor, which serves as an uncertainty estimator rather than the final classifier. The predictor consists of a single hypergraph convolution layer followed by a linear classifier and a softmax function. Given the node feature matrix $\mathbf{X}$ and the structural incidence matrix $\mathbf{H}^s$, it first produces preliminary node representations through local hypergraph propagation and then generates the preliminary class distribution

$$\mathbf{P} = [p_{ik}] \in \mathbb{R}^{n \times K},$$

where K denotes the number of classes and p_{ik} represents the probability that node v_i belongs to class k . The auxiliary predictor is trained on labeled training nodes using the standard cross-entropy loss:

$$\mathcal{L}_{aux} = - \sum_{v_i \in \mathcal{V}_L} \sum_{k=1}^K y_{ik} \log p_{ik}, \quad (5)$$

where $\mathcal{V}_L$ denotes the labeled training node set and y_{ik} is the ground-truth label indicator. The parameters of the auxiliary predictor are independent of the main DDEC model and are not shared with the dual dependency enhancement module or the final classifier. Its output $\mathbf{P}$ is used only to compute node entropy for feature recalibration and does not directly determine the final node prediction.

It should be noted that DDEC does not require the auxiliary predictor to produce well-calibrated absolute class probabilities or highly accurate final predictions. Instead, the entropy derived from $\mathbf{P}$ is used as a relative reliability ranking signal. Nodes with flatter predictive distributions are regarded as relatively more uncertain, while nodes with more confident distributions are considered more reliable. Therefore, even though the auxiliary predictor may also be affected by sparse structures, its entropy output can still provide

useful guidance for identifying relatively unreliable node signals before attribute-view construction. The entropy of node v_i is defined as

$$\mathcal{H}(v_i) = - \sum_{k=1}^K p_{ik} \log p_{ik}. \quad (6)$$

A larger entropy indicates a flatter predictive distribution and thus a more uncertain class assignment, suggesting that the corresponding node features may contain stronger ambiguity or noise. In contrast, a smaller entropy implies higher predictive confidence and greater feature stability. Therefore, entropy serves as an effective indicator for assessing the relative reliability of node features under sparse structural conditions. To explicitly incorporate uncertainty into feature modeling, we convert the entropy of each node into a reliability weight through an exponential mapping:

$$r_i = \exp(-\mathcal{H}(v_i)), \quad (7)$$

where r_i denotes the reliability score of node v_i . This exponential mapping converts entropy into a positive reliability score and assigns exponentially smaller weights to highly uncertain nodes, thereby yielding a smooth and monotonic uncertainty-aware modulation. We then construct a diagonal reliability matrix $\mathbf{R} = \text{diag}(r_1, r_2, \dots, r_n)$ and recalibrate the original node features as

$$\tilde{\mathbf{X}} = \mathbf{R}\mathbf{X}. \quad (8)$$

In this way, feature contributions are adaptively modulated according to node-specific uncertainty. Nodes with higher entropy are assigned smaller reliability weights to suppress potentially unreliable attribute signals, while nodes with lower entropy preserve more discriminative information. This node-wise recalibration reduces the adverse effect of noisy features before attribute-view construction and provides a more reliable basis for subsequent dual dependency enhancement and collaborative contrastive learning.

4.2. Dual Dependency Enhancement

To strengthen dependency modeling under sparse hypergraph structures, we perform dual dependency enhancement from both the structural and attribute views. This module consists of four stages: structural and attribute view construction, dual line-graph transformation with semantic affinity estimation, dependency enhancement with dual-space propagation, and confidence-aware multi-view fusion. Through these stages, DDEC enhances inter-hyperedge dependency modeling in the dual space and injects the enhanced relations back into node representations from two complementary views.

4.2.1. Structural and Attribute View Construction

As the first step of dual dependency enhancement, we construct two complementary hypergraph views based on the recalibrated features to characterize observed topology and latent attribute affinity, respectively. The structural view directly inherits the original incidence structure with incidence matrix $\mathbf{H}^s$ and preserves the observed node–hyperedge relations. In contrast, the attribute view is constructed from the recalibrated feature space to supplement sparse structural connectivity with feature-induced relational cues. It should be noted that the attribute view is not intended to replace the structural view, but to provide additional semantic evidence when the original node–hyperedge incidences are insufficient for reliable propagation.

Specifically, for each node v_i , we form an attribute-induced hyperedge by grouping v_i together with its k nearest neighbors according to feature similarity in the recalibrated feature space, thereby obtaining the incidence matrix $\mathbf{H}^f$. These attribute-induced hyperedges

are not intended to replace the original hypergraph structure, but to provide complementary feature-level associations under sparse topology. In this way, the attribute view introduces additional relational cues beyond the original sparse structure and supports subsequent dependency enhancement.

Nevertheless, when raw node features are extremely weak or misleading, the constructed attribute view may still contain unreliable semantic neighbors, and entropy-guided recalibration alone may not fully remove such noise. To reduce this risk, DDEC does not rely solely on the attribute view. Instead, the structural and attribute views are later integrated through a confidence-aware gating mechanism, which adaptively adjusts their contributions. When the attribute view is less reliable, its influence can be down-weighted, allowing the model to rely more on the structural view.

For each view $t \in \{s, f\}$, the recalibrated node features are projected into the corresponding hyperedge space to obtain the initial hyperedge representations:

$$\mathbf{E}^t = (\mathbf{D}_e^t)^{-1} \mathbf{W}_e^t (\mathbf{H}^t)^\top \tilde{\mathbf{X}}, \quad (9)$$

where $\mathbf{H}^t$ is the incidence matrix of view t , $\mathbf{D}_e^t$ is the corresponding hyperedge degree matrix, and $\mathbf{W}_e^t$ is the hyperedge weight matrix. Equation (9) performs a weighted average aggregation of incident node features to initialize hyperedge representations in each view.

4.2.2. Dual Line-Graph Transformation and Semantic Affinity Estimation

Based on the constructed structural and attribute views, we further enhance hyperedge dependency modeling in the dual space. For each view $t \in \{s, f\}$, we first transform the hypergraph into its dual line graph, where each hyperedge is mapped to a dual node and overlap between hyperedges is converted into explicit graph adjacency. In this way, local topological relations among hyperedges can be explicitly represented in the dual space.

However, explicit overlap-based adjacency alone is insufficient for sparse hypergraphs, since semantically related hyperedges may still be weakly connected or even disconnected in the original structure. To complement such missing relational cues, we further estimate latent semantic affinity among hyperedges through self-attention based on the initial hyperedge representations in Equation (9):

$$\mathbf{A}_{\text{att}}^t = \text{softmax} \left(\frac{Q(\mathbf{E}^t) K(\mathbf{E}^t)^\top}{\sqrt{d_k}} \right), \quad (10)$$

where $Q(\cdot)$ and $K(\cdot)$ denote the query and key projection functions, respectively, and d_k is the scaling factor.

Different from standard self-attention used for feature transformation, we retain only the normalized attention matrix as a semantic affinity estimator in the dual space. Larger values in $\mathbf{A}_{\text{att}}^t$ indicate stronger latent semantic correlations among hyperedges. As a result, each dual space contains two complementary types of relations: explicit topological adjacency induced by the line graph and implicit semantic affinity estimated from hyperedge representations.

4.2.3. Dependency Enhancement and Dual-Space Propagation

After obtaining explicit dual topological adjacency and attention-based semantic affinity, we integrate them to enhance dependency modeling in sparse hypergraphs. Specifically, the enhanced dual adjacency matrix is constructed as

$$\tilde{\mathbf{A}}^t = \alpha \mathbf{A}_{\text{top}}^t + (1 - \alpha) \text{TopK}_s(\mathbf{A}_{\text{att}}^t), \quad (11)$$

where $\mathbf{A}_{\text{top}}^t$ denotes the topological adjacency induced by the dual line graph, α is a balancing coefficient, and $\text{TopK}_s(\cdot)$ retains the largest s entries in each row while setting the remaining entries to zero. This sparsification operation filters weak and potentially noisy semantic affinities, while preserving the most informative inter-hyperedge relations for sparse dependency enhancement.

The above fusion preserves explicit topological relations in the dual graph while supplementing them with attention-based semantic connections. Consequently, hyperedges that are weakly connected in topology but semantically related can still interact in the enhanced dual space, which facilitates dependency modeling beyond local structural overlap.

Given the enhanced dual adjacency matrix, we update hyperedge representations in the dual space via graph convolution:

$$\mathbf{Z}_e^t = \sigma\left((\tilde{\mathbf{D}}^t)^{-\frac{1}{2}} \tilde{\mathbf{A}}^t (\tilde{\mathbf{D}}^t)^{-\frac{1}{2}} \mathbf{E}^t \Theta_e^t\right), \quad (12)$$

where $\tilde{\mathbf{D}}^t$ is the degree matrix of $\tilde{\mathbf{A}}^t$, Θ_e^t is a learnable parameter matrix, and $\sigma(\cdot)$ is the nonlinear activation function. Since the dual line graph treats hyperedges as nodes, graph convolution in the dual space enables explicit propagation of inter-hyperedge dependencies that cannot be directly captured in the original incidence space.

The enhanced hyperedge representations are then projected back to the node space to obtain view-specific node representations:

$$\mathbf{Z}_v^t = \sigma\left((\mathbf{D}_v^t)^{-1} \mathbf{H}^t \mathbf{Z}_e^t \Theta_v^t\right), \quad (13)$$

where $\mathbf{D}_v^t$ is the node degree matrix of view t and Θ_v^t is the projection parameter matrix. Through this projection, the enhanced dependencies learned in the dual space are injected back into node representations for each view.

4.2.4. Confidence-Aware Multi-View Fusion

After dual dependency enhancement in the structural and attribute views, we obtain view-specific node representations $\mathbf{Z}_v^s$ and $\mathbf{Z}_v^f$. Since these two views provide complementary information, we further design a confidence-aware gated fusion mechanism to adaptively integrate them. Different from the entropy-based reliability used for input recalibration, the confidence defined here is view-specific and is used to estimate the predictive quality of each view for adaptive fusion.

First, the node representations from each view are mapped to the label space to obtain the corresponding predictive probability matrices:

$$\mathbf{P}^t = \text{softmax}(\mathbf{Z}_v^t \mathbf{W}_c + \mathbf{b}), \quad t \in \{s, f\}, \quad (14)$$

where $\mathbf{W}_c$ is a shared learnable mapping matrix, $\mathbf{b}$ is a bias term, and the number of classes is K . The shared mapping projects both views into the same label space, making their prediction scores comparable for confidence estimation.

To assess the reliability of each view at the prediction level, we define a confidence vector by jointly considering the maximum class probability and the margin between the largest and second-largest probabilities:

$$\mathbf{c}^t = \max(\mathbf{P}^t) + (\max(\mathbf{P}^t) - 2\text{ndmax}(\mathbf{P}^t)), \quad (15)$$

where $\max(\mathbf{P}^t)$ and $2\text{ndmax}(\mathbf{P}^t)$ denote the row-wise largest and second-largest values of $\mathbf{P}^t$, respectively. This definition jointly reflects absolute prediction confidence and relative class separation, thereby providing a simple yet effective estimate of view-specific reliability.

The confidence scores from the two views are normalized to obtain the corresponding gating weights:

$$\mathbf{G}^t = \frac{\mathbf{c}^t}{\mathbf{c}^s + \mathbf{c}^f}, \quad t \in \{s, f\}. \quad (16)$$

The resulting gating weights are node-wise scalar coefficients, which are broadcast along the feature dimension to modulate the contribution of each view. Based on these weights, the final fused node representation is given by

$$\mathbf{Z} = \mathbf{G}^s \odot \mathbf{Z}_v^s + \mathbf{G}^f \odot \mathbf{Z}_v^f, \quad (17)$$

where $\odot$ denotes the Hadamard product.

Finally, the fused node representation is fed into a two-layer multilayer perceptron followed by a softmax classifier to produce the final prediction:

$$\hat{\mathbf{Y}} = \text{softmax}(\text{MLP}(\mathbf{Z})). \quad (18)$$

4.3. Multi-Granularity Collaborative Contrastive Learning

Although the preceding modules provide more reliable node representations and enhanced dependency modeling, supervised classification alone is still insufficient to constrain semantic consistency across different granularities in sparse hypergraphs. In particular, sparse topology may weaken local contextual coherence at the node level and impair the stability of global dependency semantics at the hyperedge level. To address this issue, we introduce a multi-granularity collaborative contrastive learning strategy that imposes complementary self-supervised constraints on both nodes and hyperedges. In this way, local semantic consistency and global dependency consistency are jointly preserved to further improve the robustness and discriminability of the learned representations.

4.3.1. Node-Level Contrastive Loss

To preserve local semantic consistency under sparse structural conditions, we first introduce a node-level contrastive objective on the fused node representations. Let $\mathbf{Z}$ denote the fused node representation matrix. For a node v_i , we define its positive sample set $\mathcal{P}_i$ as the set of nodes that share at least one hyperedge with v_i , and its negative sample set $\mathcal{N}_i$ as the set of nodes that have no direct hyperedge association with v_i . Based on the InfoNCE formulation, the node-level contrastive loss is defined as

$$\mathcal{L}_{\text{node}} = - \sum_{v_i \in \mathcal{V}} \log \frac{\sum_{v_j \in \mathcal{P}_i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j) / \tau)}{\sum_{v_j \in \mathcal{P}_i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j) / \tau) + \sum_{v_k \in \mathcal{N}_i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k) / \tau)}, \quad (19)$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is the temperature coefficient. This objective encourages nodes sharing hyperedge context to remain close in the embedding space, thereby strengthening local semantic coherence and improving the stability of node representations under sparse topology.

4.3.2. Hyperedge-Level Contrastive Loss

Beyond node-level local consistency, sparse hypergraph learning also requires preserving dependency-aware semantic relations among hyperedges. After dual dependency enhancement, hyperedges that are strongly connected in the enhanced dual space should remain close in the representation space. To this end, we further introduce a hyperedge-level contrastive objective to stabilize global dependency semantics.

Let $\mathbf{Z}_e^t$ denote the hyperedge representation matrix in the dual space for view $t \in \{s, f\}$. For each view, two hyperedges are treated as a positive pair if they are connected in

the enhanced dual adjacency matrix $\tilde{\mathbf{A}}^t$, and as a negative pair otherwise. Accordingly, the hyperedge-level contrastive loss is defined as

$$\mathcal{L}_{\text{edge}} = - \sum_{e_i \in \mathcal{E}} \log \frac{\sum_{e_j \in \mathcal{P}_i^e} \exp(\text{sim}(\mathbf{e}_i, \mathbf{e}_j) / \tau)}{\sum_{e_j \in \mathcal{P}_i^e} \exp(\text{sim}(\mathbf{e}_i, \mathbf{e}_j) / \tau) + \sum_{e_k \in \mathcal{N}_i^e} \exp(\text{sim}(\mathbf{e}_i, \mathbf{e}_k) / \tau)}, \quad (20)$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, and $\mathcal{P}_i^e$ and $\mathcal{N}_i^e$ denote the positive and negative hyperedge sets associated with hyperedge e_i , respectively. This objective encourages hyperedges with strong topological or semantic dependencies in the enhanced dual space to stay close in the embedding space, thereby preserving global dependency consistency and reducing semantic drift caused by sparse structural fragmentation.

4.3.3. Joint Optimization Objective

We jointly optimize the supervised classification objective together with the two contrastive objectives. The overall loss function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_{\text{node}} + \lambda_2 \mathcal{L}_{\text{edge}}, \quad (21)$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy classification loss, and λ_1 and λ_2 are hyperparameters that control the strengths of the node-level and hyperedge-level contrastive objectives, respectively.

Through joint optimization, the node-level contrastive objective preserves local semantic consistency, while the hyperedge-level contrastive objective stabilizes global dependency semantics. Together, they complement supervised classification and further improve the robustness and discriminability of representations learned on sparse hypergraphs.

4.4. Algorithmic Procedure

The overall procedure of DDEC is summarized in Algorithm 1. Given the original hypergraph and node feature matrix, DDEC first performs entropy-guided feature recalibration to obtain more reliable node features. Based on the recalibrated features, it then constructs structural and attribute views, enhances hyperedge dependencies in the dual space for each view, and adaptively fuses the resulting node representations through confidence-aware gating. Finally, node-level and hyperedge-level contrastive objectives are jointly optimized together with the classification loss to learn robust and discriminative representations for sparse hypergraph node classification.

4.5. Time Complexity Analysis

To analyze the computational efficiency of DDEC, we summarize the time complexity of its main components. Let n denote the number of nodes, d the hidden dimension, c the number of classes, and m^s and m^f the numbers of hyperedges in the structural and attribute views, respectively. Let $|\mathcal{I}^s|$ and $|\mathcal{I}^f|$ denote the numbers of node–hyperedge incidences in the two views. We use q to denote the number of candidate hyperedge neighbors used for semantic affinity estimation and s to denote the number of retained neighbors after sparsification.

The entropy-guided feature recalibration module obtains preliminary predictions through a lightweight auxiliary hypergraph predictor, with complexity $O(|\mathcal{I}^s|d + ndc)$. Entropy estimation and feature recalibration further require $O(nc + nd)$. The structural view is directly inherited from the original incidence matrix without extra construction cost. The attribute view is constructed once from recalibrated node features using a k -nearest-neighbor strategy, which costs $O(n^2d)$ in a straightforward implementation. Since this step is performed only once before training, it is treated as preprocessing and excluded from the per-epoch complexity.

Algorithm 1: The overall procedure of DDEC

Input: Hypergraph $\mathcal{G}$, node feature matrix $\mathbf{X}$, balancing coefficient α , neighborhood size k , sparsification parameter s , temperature τ , loss weights λ_1 and λ_2

Output: Predicted node labels

- 1 Obtain the preliminary predictive distribution $\mathbf{P}$ using an auxiliary hypergraph predictor;
- 2 Compute node entropy $\mathcal{H}(v_i)$ according to Equation (6);
- 3 Construct the reliability matrix $\mathbf{R}$ and recalibrate node features $\tilde{\mathbf{X}}$ via Equation (8);
- 4 Inherit the structural view $\mathbf{H}^s$ from the original incidence matrix and construct the attribute view $\mathbf{H}^f$ based on $\tilde{\mathbf{X}}$; Project $\tilde{\mathbf{X}}$ into the hyperedge spaces of the two views to obtain initial hyperedge representations via Equation (9);
- 5 For each view, transform the hypergraph into its dual line graph and construct the dual topological adjacency matrix;
- 6 Estimate the hyperedge semantic affinity matrix according to Equation (10);
- 7 Fuse explicit topology and semantic affinity to obtain the enhanced dual adjacency matrix via Equation (11);
- 8 Update hyperedge representations in the dual space using Equation (12);
- 9 Project the enhanced hyperedge representations back to the node space via Equation (13);
- 10 Compute the confidence scores and gating weights according to Equations (15) and (16), and fuse the two-view node representations via Equation (17);
- 11 Compute the node-level contrastive loss $\mathcal{L}_{\text{node}}$ and the hyperedge-level contrastive loss $\mathcal{L}_{\text{edge}}$ according to Equations (19) and (20);
- 12 Optimize the overall objective in Equation (21) and update model parameters;
- 13 Output the final node predictions according to Equation (18);

For each view $t \in \{s, f\}$, projecting node representations into the hyperedge space and projecting enhanced hyperedge representations back to the node space cost $O(|\mathcal{I}^t|d)$. In dual dependency enhancement, semantic affinity is estimated only within a candidate neighbor set of size q , rather than by dense all-pair computation, leading to complexity $O(m^t q d)$. After top- s sparsification, dual-space graph convolution is performed on the sparse enhanced adjacency matrix with complexity $O(m^t s d)$. The confidence-aware two-view fusion costs $O(nd)$.

For optimization, the supervised classification loss costs $O(nc)$. The node-level contrastive loss is computed over sparse incidence-induced positive pairs, with complexity $O((|\mathcal{I}^s| + |\mathcal{I}^f|)d)$. The hyperedge-level contrastive loss is computed over retained dual-space neighbors, with complexity $O((m^s s + m^f s)d)$.

Therefore, excluding the one-time attribute-view preprocessing cost, the per-epoch time complexity of DDEC can be summarized as

$$O\left(ndc + \sum_{t \in \{s, f\}} (|\mathcal{I}^t|d + m^t q d + m^t s d)\right). \quad (22)$$

Since q and s are much smaller than the number of hyperedges in sparse hypergraphs, DDEC avoids dense hyperedge-level dependency computation and mainly operates on sparse incidence matrices and sparse dual-space adjacency matrices.

5. Experimental Results and Analysis

5.1. Experimental Settings

To comprehensively evaluate the effectiveness of DDEC for sparse hypergraph node classification, we conduct experiments on eight public datasets drawn from diverse real-world scenarios, including academic networks, biological data, social networks, and e-commerce/multimedia interaction networks. These datasets exhibit substantial diversity in terms of structural sparsity, feature dimensionality, and class cardinality, thereby providing a comprehensive test bed for assessing the generalizability and robustness of the proposed framework across different sparse hypergraph scenarios. The detailed statistics of all datasets are reported in Table 1.

Table 1. Statistics of the Datasets

Dataset	n	d	m^s	m^f	ρ_e	ρ_v	ζ	K	h
Cora-C	2708	1433	1579	2708	1.77	3.03	1.25	7	0.8625
Citeseer	3312	3703	1079	3312	1.04	3.20	0.85	6	0.8291
Cora-A	2708	1433	1072	2708	1.69	4.28	0.75	7	0.8763
Zoo	101	16	43	101	17	39.93	5.58	7	0.6558
Mushroom	8124	22	298	8124	5	136.31	0.09	2	0.9554
Actor	16,255	50	10,164	16,255	3.39	5.43	0.31	3	0.6755
Amazon	22,299	111	2090	22,299	0.29	3.10	1.60	5	0.6441
Pokec	14,998	65	2406	14,998	0.37	2.29	0.66	2	0.7225

Note: n denotes the number of nodes, d denotes the dimensionality of node features, m^s and m^f denote the numbers of hyperedges in the structural-view and attribute-view hypergraphs, respectively, ρ_e denotes the average hyperedge cardinality, ρ_v denotes the average node participation, ζ denotes the average number of neighboring hyperedges in the dual space, K denotes the number of classes, and h denotes the intra-hyperedge label homophily ratio, which measures the proportion of node pairs within the same hyperedge that share the same class label.

To verify the effectiveness of the proposed framework, we compare DDEC against 10 representative hypergraph learning baselines spanning three categories: classical hypergraph neural networks, dynamic structure modeling methods, and contrastive-learning-based hypergraph representation learning methods. Specifically, the classical baselines include HGNN, HNHN, and AllSet; the dynamic structure modeling baselines include DHGNN and TDHNN, and the recent hypergraph method DPHGNN, which we include as an additional baseline using hyperedges reconstructed from our datasets to ensure fair comparison; the contrastive baselines include HyperGCL, HSL, CHGNN, HyGCL-AdT, and we note that HyperGT is not included because its original experimental setting focuses on featureless hypergraphs, which differs from our attributed sparse hypergraph node classification scenario and would require nontrivial adaptation. This broad selection enables a comprehensive evaluation of DDEC against different families of competitive methods.

We adopt classification accuracy (ACC) as the evaluation metric. For each dataset, the training, validation, and test sets are split according to a ratio of 1:1:8. To ensure a fair comparison, all baseline methods are trained and evaluated under the same data split.

Unless otherwise specified, the key hyperparameters of DDEC are set as follows. The number of nearest neighbors used to construct the attribute-view hypergraph is fixed at $k = 10$. In the dual dependency enhancement module, the fusion coefficient between explicit topological adjacency and attention-derived semantic affinity is set to $\alpha = 0.6$, and the sparsification threshold for semantic affinity is set to $s = 5$. In the joint optimization objective, the weights of the node-level and hyperedge-level contrastive losses are set to $\lambda_1 = 0.3$ and $\lambda_2 = 0.4$, respectively.

For the entropy-guided feature recalibration module, the auxiliary predictor is a lightweight hypergraph convolutional network consisting of a single hypergraph convolution layer followed by a linear classifier and softmax function to generate preliminary class

distributions $\mathbf{P}$. It is trained on the same training split using standard cross-entropy loss. The parameters of the auxiliary predictor are independent of the main DDEC model and are not shared with other modules. Its output is only used to compute node-level predictive entropy to guide feature recalibration; it does not directly participate in final prediction. Therefore, the auxiliary predictor serves as a fixed uncertainty estimator rather than an additional prediction branch in the main model. Its contribution to final performance is confirmed in the ablation study in Section 5.3.

All experiments are implemented in PyTorch 2.3.1 and conducted on a single NVIDIA A100 GPU. To reduce the influence of randomness, each experiment is repeated five times with different random seeds, and the reported results are averaged over these runs. Baselines follow the parameter settings recommended in their original papers and are further tuned on validation sets to ensure fair comparison.

5.2. Overall Performance Comparison

Table 2 reports the node classification accuracy of DDEC and all baseline methods on eight public datasets. Overall, DDEC achieves superior or competitive performance across all datasets, obtaining the best results on five datasets: Cora-C, Citeseer, Cora-A, Zoo, and Amazon. These results demonstrate the effectiveness of DDEC for sparse hypergraph node classification across diverse real-world scenarios.

Table 2. Comparison of classification accuracy (%) between DDEC and baseline methods

Method	Cora-C	Citeseer	Cora-A	Zoo	Mushroom	Amazon	Pokec	Actor
HGNN	62.99	56.68	70.74	77.78	99.01	25.50	57.34	73.57
HNHN	76.21	67.28	74.88	74.04	97.00	26.34	56.33	71.76
AllSet	76.21	67.83	76.94	65.43	98.48	25.63	56.04	77.12
DHGNN	70.88	63.92	75.91	79.01	OOM	24.31	54.73	63.47
HSL	75.45	65.43	74.17	76.54	<u>99.69</u>	26.12	50.88	82.97
TDHNN	68.76	66.64	71.94	41.98	97.09	<u>27.02</u>	57.02	62.46
HyperGCL	73.52	66.82	76.57	66.89	99.46	26.09	53.80	75.44
CHGNN	<u>77.00</u>	68.96	79.95	77.38	98.89	26.44	50.78	66.09
HyGCL-AdT	76.64	<u>70.45</u>	<u>80.25</u>	76.54	99.77	25.87	58.68	<u>81.37</u>
DPHGNN	49.05	40.20	73.37	<u>80.00</u>	94.83	19.24	55.53	64.27
DDEC	78.31	72.87	81.17	81.48	99.54	27.40	<u>57.64</u>	80.97
Improve	1.31	2.42	0.92	1.48	−0.23	0.38	−1.04	−2.00

Note: The best results are shown in bold, and the second-best results are underlined. “OOM” indicates that the method runs out of memory. “Improve” denotes the absolute performance gain of DDEC over the strongest baseline on each dataset.

More specifically, DDEC outperforms the strongest competing baseline by 1.31, 2.42, 0.92, 1.48, and 0.38 percentage points on Cora-C, Citeseer, Cora-A, Zoo, and Amazon, respectively. The gains on Citeseer, Zoo, and Cora-C are particularly notable. According to the statistics in Table 1, Cora-C, Citeseer, and Cora-A have relatively high intra-hyperedge label homophily ratios, with $h = 0.8625$, 0.8291 , and 0.8763 , respectively. This indicates that nodes within the same hyperedge tend to share relatively coherent class semantics, making node-level consistency modeling more reliable. In addition, Zoo has a much larger average number of neighboring hyperedges in the dual space, with $\zeta = 5.58$, suggesting that rich inter-hyperedge dependencies remain recoverable even though its intra-hyperedge homophily is not very high. These observations indicate that DDEC can benefit from either reliable local semantic consistency or recoverable global dependency structures, especially when sparse high-order interactions are only weakly observed from the original topology.

On Mushroom, DDEC performs slightly worse than the best baseline. Although Mushroom has a high intra-hyperedge label homophily ratio ($h = 0.9554$), its average number of neighboring hyperedges in the dual space is extremely small ($\zeta = 0.09$), indicating that

there are very limited inter-hyperedge associations to recover. Under such conditions, the effectiveness of dual dependency enhancement is constrained, and the relational supervision provided by the contrastive objectives is also weakened. This explains why DDEC only achieves limited improvement on this dataset despite its strong local homophily.

DDEC also does not achieve the best result on Pokec and Actor. From Table 1, both datasets have relatively low dual-space connectivity and weaker intra-hyperedge semantic consistency. Specifically, Pokec has $\zeta = 0.66$ and $h = 0.7225$, while Actor has $\zeta = 0.31$ and $h = 0.6755$. These statistics indicate that the recoverable inter-hyperedge dependencies are limited and that nodes within the same hyperedge may not share highly consistent class semantics. For example, in Pokec, a hyperedge corresponds to a friendship group whose members may differ in gender, interests, and other attributes. In Actor, a hyperedge corresponds to a movie project and may include actors, directors, and screenwriters with different roles. In such cases, enforcing node-level local consistency may introduce ambiguous supervision and blur the original class boundaries, thereby reducing classification performance.

Overall, the results suggest that DDEC is most effective when useful inter-hyperedge dependencies remain recoverable in the dual space, and relatively high intra-hyperedge homophily can further enhance the benefit of node-level consistency modeling. When dual-space connectivity is extremely weak, as in Mushroom, the benefit of dual dependency enhancement becomes limited even if local homophily is high. When both dual-space connectivity and intra-hyperedge semantic consistency are relatively weak, as in Pokec and Actor, the advantage of DDEC is also reduced. These observations provide more concrete guidance for applying DDEC: it is particularly suitable for sparse hypergraphs with recoverable global dependency structures, while reliable local contextual consistency can further improve its stability.

5.3. Ablation Study

To investigate the contribution of each component in DDEC, we conduct ablation studies on Cora-A, Cora-C, and Citeseer. Six model variants are constructed by removing one component at a time, including feature recalibration, dual line-graph transformation, attention-based semantic enhancement, node-level contrastive learning, hyperedge-level contrastive learning, and the attribute view. The variant definitions are listed in Table 3, and the corresponding results are shown in Figure 3.

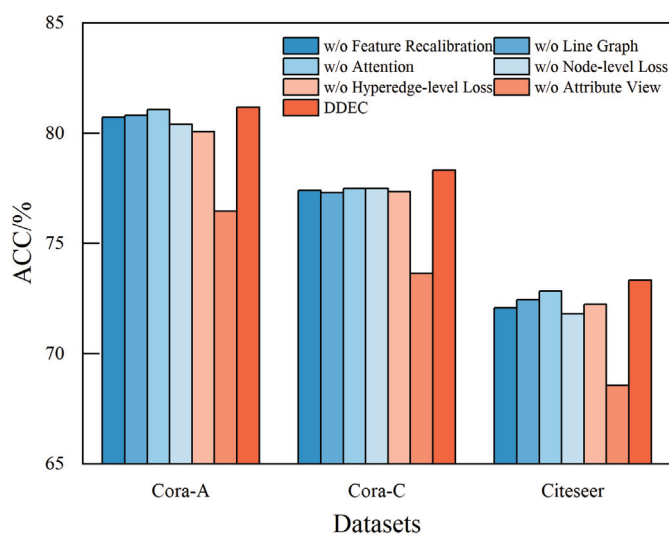


Figure 3. Ablation study of the core components in DDEC for sparse hypergraph node classification.

Overall, the full DDEC model achieves the best performance on all three datasets, which demonstrates that the proposed components contribute to sparse hypergraph node classification in a highly complementary manner. In contrast, removing any single component leads to a noticeable performance drop, indicating that uncertainty-aware feature recalibration, dual dependency enhancement, and multi-granularity collaborative contrastive learning all play important roles in the final performance gains.

Table 3. Variants used in the ablation study of DDEC.

Variant	Feature Recalibration	Line Graph	Attention	Node-Level Loss	Hyperedge-Level Loss	Attribute View
w/o Feature Recalibration	×	✓	✓	✓	✓	✓
w/o Line Graph	✓	×	✓	✓	✓	✓
w/o Attention	✓	✓	×	✓	✓	✓
w/o Node-level Loss	✓	✓	✓	×	✓	✓
w/o Hyperedge-level Loss	✓	✓	✓	✓	×	✓
w/o Attribute View	✓	✓	✓	✓	✓	×
DDEC	✓	✓	✓	✓	✓	✓

Note: ✓ indicates that the corresponding module is used, while × indicates that it is removed.

Specifically, removing the feature recalibration module decreases accuracy by 0.46, 0.92, and 1.24 percentage points on Cora-A, Cora-C, and Citeseer, respectively. This module relies on the lightweight auxiliary hypergraph predictor, which is trained independently and used to generate preliminary class distributions for computing node-level entropy. The entropy serves as a relative reliability signal, suppressing unreliable node features before constructing the attribute view. The observed performance drop confirms that this entropy-guided feature recalibration effectively mitigates the impact of noisy or ambiguous node signals, providing more stable inputs for subsequent dependency modeling.

For the dual dependency enhancement module, removing the line graph, attention mechanism, or attribute view consistently degrades performance, indicating that jointly modeling explicit topological adjacency and latent semantic affinity is critical for recovering incomplete inter-hyperedge dependencies in sparse hypergraphs. Among these variants, removing the attribute view causes the largest drop, with accuracy decreasing by 4.70, 4.66, and 4.75 percentage points on the three datasets, respectively. This suggests that the attribute view provides crucial supplementary relational cues beyond the sparse structural view, especially when high-order interactions are only weakly observed from topology alone.

For the collaborative contrastive learning module, removing the node-level contrastive loss results in accuracy drops of 0.78, 0.83, and 1.51 percentage points, while removing the hyperedge-level contrastive loss leads to decreases of 1.11, 0.97, and 1.09 percentage points on the three datasets, respectively. These results indicate that the node-level contrastive objective helps preserve local semantic consistency among structurally related nodes, whereas the hyperedge-level contrastive objective stabilizes global dependency semantics in the dual space. Together, the two objectives provide complementary supervision from local node relations and global hyperedge dependencies, thereby improving the discriminability and stability of the learned representations.

In summary, the ablation results show that the performance gains of DDEC do not arise from any single design choice, but from the joint effect of uncertainty-aware feature recalibration, dual dependency enhancement, and multi-granularity collaborative contrastive learning. In particular, the attribute view and dual-space dependency enhancement help recover incomplete inter-hyperedge dependencies under sparse topology, while node-level and hyperedge-level contrastive learning further preserve local semantic consistency and stabilize global dependency semantics.

5.4. Parameter Sensitivity Analysis

To further investigate the influence of key hyperparameters, we conduct a sensitivity analysis on the Citeseer dataset. Specifically, we examine the fusion coefficient α and the sparsification threshold s in the dual dependency enhancement module, as well as the loss weights λ_1 and λ_2 in the joint optimization objective.

Figure 4a illustrates the classification accuracy under different combinations of α and s , where $s \in [0, 20]$. The results show that model performance is sensitive to both hyperparameters. When α is too large, the model relies excessively on explicit topological adjacency and fails to fully exploit latent semantic affinity among hyperedges. In contrast, when α is too small, the structural constraints inherited from the original hypergraph become insufficient. Similarly, when s is too small, only a limited number of meaningful semantic connections are retained. When s is too large, weak and noisy relations are more likely to be introduced. The best performance is achieved at $\alpha = 0.6$ and $s = 5$, indicating that explicit topological adjacency and attention-derived semantic affinity should be properly balanced, and that retaining only a small number of high-confidence semantic links is most beneficial for dual-space dependency enhancement.

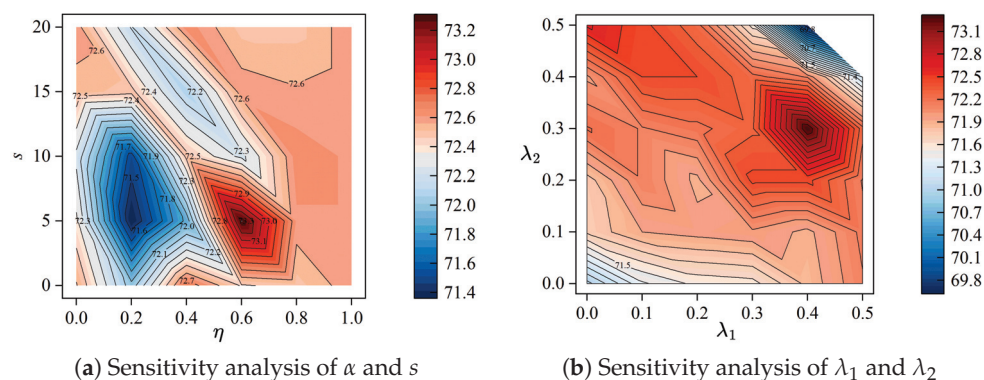


Figure 4. Sensitivity analysis of key parameters.

Figure 4b presents the heat map of classification accuracy under different values of λ_1 and λ_2 . When both weights are too small, the node-level and hyperedge-level contrastive losses are too weak to provide effective auxiliary supervision for representation learning. In contrast, when both are too large, the contrastive objectives dominate the optimization process and weaken the contribution of the supervised classification signal, leading to performance degradation. The best result is obtained when $\lambda_1 = 0.3$ and $\lambda_2 = 0.4$. This suggests that the node-level and hyperedge-level contrastive objectives should be balanced appropriately with the classification objective in order to fully exploit their complementary roles in preserving local semantic consistency and stabilizing global dependency semantics.

5.5. Robustness Analysis

To evaluate the robustness of DDEC under sparse perturbations, we conduct robustness experiments on Cora-A, Cora-C, and Citeseer from two perspectives: structural perturbation and feature perturbation. DDEC is compared with representative methods, including HGNN and CHGNN, and classification accuracy is used as the evaluation metric. In the structural robustness experiments, 40%, 50%, and 60% of node–hyperedge incidence relations are randomly removed, followed by the deletion of 70% of hyperedges to simulate different levels of structural damage. In the feature robustness experiments, partial node features are randomly masked and set to zero, with masking ratios $p_{\text{mask}} \in \{20\%, 30\%, 40\%\}$. The results are shown in Figures 5 and 6, respectively.

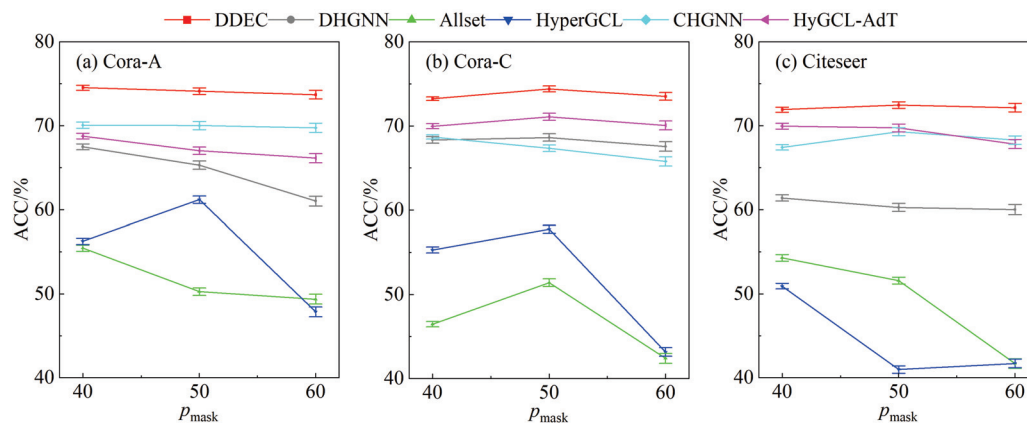


Figure 5. Robustness comparison of DDEC and baseline methods under structural perturbation.

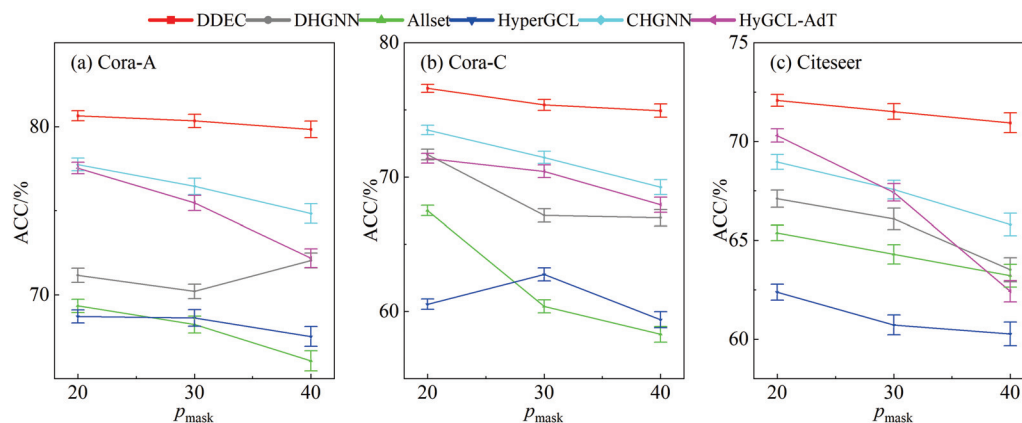


Figure 6. Robustness comparison of DDEC and baseline methods under feature perturbation.

As shown in Figure 5, although the performance of all methods fluctuates as structural perturbation becomes stronger, DDEC consistently remains superior on Cora-A, Cora-C, and Citeseer. This demonstrates that DDEC has strong robustness under structural corruption. Such robustness can be attributed to three factors. First, uncertainty-aware feature recalibration suppresses unreliable signal propagation. Second, dual dependency enhancement helps recover incomplete inter-hyperedge dependencies under sparse topology. Third, collaborative contrastive learning further preserves node-level local semantic consistency and stabilizes hyperedge-level global dependency semantics. As a result, DDEC remains stable even under severely corrupted structural conditions.

Figure 6 further shows that, as feature perturbation increases, the classification accuracy of all methods generally declines. Nevertheless, DDEC still outperforms the competing baselines on all three datasets, indicating that it is also robust to feature corruption. This result suggests that the proposed feature recalibration effectively alleviates the interference caused by low-confidence features, while the attribute view and confidence-aware multi-view fusion help compensate for missing or corrupted node attributes by exploiting complementary feature-induced relational cues. Consequently, DDEC maintains strong classification capability even when the input features are partially damaged.

5.6. Training Time Comparison

To further evaluate the practical efficiency of DDEC, we report the wall-clock training time of DDEC and two representative baselines, HyGCL-AdT and DPHGNN, on five datasets. All methods are evaluated under the same experimental environment, and the results are shown in Table 4.

Table 4. Training time (in seconds) of DDEC and representative baselines on five datasets.

Method	Cora-C	Citeseer	Cora-A	Zoo	Mushroom
HyGCL-AdT	60	72	86	31	282.83
DPHGNN	182	220	168	129	4620
DDEC	206	227	123	12	5580

As shown in Table 4, DDEC generally requires more training time than simpler baselines on large-scale datasets. This is mainly because DDEC introduces dual line-graph transformation, hyperedge-level semantic affinity estimation, and node-level and hyperedge-level contrastive learning. Nevertheless, the additional cost is expected and remains manageable on modern GPUs. Moreover, DDEC can be successfully trained on all datasets, indicating its practical feasibility for sparse hypergraph node classification.

6. Conclusions and Future Work

In this paper, we propose DDEC, a Dual Dependency-Enhanced Contrastive Learning framework for sparse hypergraph node classification. DDEC aims to recover latent high-order interactions from sparse and noisy observations by jointly leveraging structural and attribute information. Specifically, it introduces an attribute view to complement missing relational information, enhances feature reliability via entropy-guided recalibration, strengthens dual dependencies in a shared relational space through line-graph transformation, and preserves multi-granularity semantic consistency via collaborative contrastive learning at both node and hyperedge levels. Extensive experiments on eight public datasets show that DDEC achieves strong performance on most datasets while exhibiting robust behavior under structural and feature perturbations, although its improvements are limited on some low-connectivity or heterogeneous datasets.

Despite its promising performance, the proposed method still has several limitations that deserve further investigation. First, the current framework relies, to some extent, on the homophily assumption within hyperedges, and its adaptability to strongly heterogeneous hypergraphs remains to be improved. Second, as the scale of the hypergraph increases, dual-space modeling and attention computation may introduce considerable computational overhead. Future work may therefore explore more efficient strategies, such as subgraph sampling, sparse attention, or approximate propagation, to improve scalability. Finally, the present study focuses on static hypergraphs. Extending dual dependency enhancement and collaborative contrastive learning to dynamic hypergraph settings, in order to capture the temporal evolution of high-order relations, is also a promising direction for future research.

Author Contributions: Conceptualization, M.L., S.Y. and W.Z.; methodology, M.L. and S.Y.; software, S.Y.; validation, M.L. and S.Y.; formal analysis, M.L.; writing—original draft preparation, M.L.; writing—review and editing, M.L. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62072292).

Data Availability Statement: The data used in this study are publicly available online or are available from the corresponding authors.

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

References

1. Gavin, A.C.; Bösch, M.; Krause, R.; Grandi, P.; Marzioch, M.; Bauer, A.; Schultz, J.; Rick, J.M.; Michon, A.M.; Cruciat, C.-M.; et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **2002**, *415*, 141–147. [CrossRef]
2. Zheng, W.P.; Liu, M.L.; Liang, J.Y. Hypergraphs: Concepts, applications and analysis. In Proceedings of the 2022 IEEE 13th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP 2022), Beijing, China, 25–27 November 2022; pp. 1–6.
3. Yin, K.Z.; Jing, X.L.; Yang, G.K.; Chu, J. Mitigating impacts of hyperedge heterogeneity on semi-supervised hypergraph contrastive learning. *Neurocomputing* **2025**, *650*, 130863. [CrossRef]
4. Ji, S.Y.; Wei, Y.X.; Dai, Q.H.; Gao, Y. Efficient hypergraph neural network on million-level data. *Sci. Sin. Inf.* **2024**, *54*, 853–871. (In Chinese)
5. Han, J.D.; Tao, Q.; Tang, Y.F.; Xia, Y. DH-HGCN: Dual homogeneity hypergraph convolutional network for multiple social recommendations. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain, 11–15 July 2022; pp. 2190–2194.
6. Ko, Y.Y.; Tong, H.H.; Kim, S.W. Enhancing hyperedge prediction with context-aware self-supervised learning. *IEEE Trans. Knowl. Data Eng.* **2025**, *37*, 1772–1784.
7. Feng, Y.F.; You, H.X.; Zhang, Z.Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019), Honolulu, HI, USA, 27 January–1 February 2019; pp. 3558–3565.
8. Gao, Y.; Feng, Y.F.; Ji, S.Y.; Ji, R. HGNN+: General hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3181–3199. [PubMed]
9. Dong, Y.H.; Sawin, W.; Bengio, Y. HNH: Hypergraph networks with hyperedge neurons. In Proceedings of the Graph Representation Learning and Beyond Workshop at the International Conference on Machine Learning (ICML 2020), Virtual, 13–18 July 2020.
10. Chien, E.; Pan, C.; Peng, J.; Milenkovic, O. You are AllSet: A multiset function framework for hypergraph neural networks. In Proceedings of the 10th International Conference on Learning Representations (ICLR 2022), Virtual, 25–29 April 2022.
11. Bai, S.; Zhang, F.; Torr, P.H.S. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* **2021**, *110*, 107637. [CrossRef]
12. Huang, J.; Yang, J. UniGNN: A unified framework for graph and hypergraph neural networks. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021), Montreal, QC, Canada, 19–27 August 2021; pp. 2563–2569.
13. Jiang, J.W.; Wei, Y.X.; Feng, Y.F.; Cao, J.; Gao, Y. Dynamic hypergraph neural networks. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), Macao, China, 10–16 August 2019; pp. 2635–2641.
14. Kang, X.; Li, X.; Yao, H.; Li, D.; Jiang, B.; Peng, X.; Wu, T.; Qi, S.; Dong, L. Dynamic hypergraph neural networks based on key hyperedges. *Inf. Sci.* **2022**, *616*, 37–51. [CrossRef]
15. Zhou, P.; Wu, Z.Q.; Zeng, X.X.; Wen, G.; Ma, J.; Zhu, X. Totally dynamic hypergraph neural networks. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI 2023), Macao, China, 19–25 August 2023; pp. 2476–2483.
16. Zhang, J.Y.; Chen, Y.Z.; Xiao, X.; Lu, R.; Xia, S.T. Learnable hypergraph Laplacian for hypergraph learning. In Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022), Singapore, 22–27 May 2022; pp. 4503–4507.
17. Lei, F.Y.; Huang, J.H.; Jiang, J.J.; Huang, D.; Li, Z.; Wang, C.D. Unveiling the potential of long-range dependence with mask-guided structure learning for hypergraph. *Knowl.-Based Syst.* **2024**, *284*, 111254.
18. Saxena, S.; Ghatak, S.; Kolla, R.; Mukherjee, D.; Chakraborty, T. DPHGNN: A dual perspective hypergraph neural networks. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2024), Barcelona, Spain, 25–29 August 2024; pp. 2548–2559.
19. Saifuddin, K.M.; Aktas, M.E.; Akbas, E. Topology-guided hypergraph transformer network: Unveiling structural insights for improved representation. *arXiv* **2023**, arXiv:2310.09657.
20. Ye, R.P.; Pei, X.B.; Yang, H.R.; Wang, R. Hyperedge interaction-aware hypergraph neural network. *arXiv* **2024**, arXiv:2401.15587.
21. Bandyopadhyay, S.; Das, K.; Murty, M.N. Line hypergraph convolution network: Applying graph convolution for hypergraphs. *arXiv* **2020**, arXiv:2002.03392.
22. Liu, Z.; Tang, B.; Ye, Z.; Dong, X.; Chen, S.; Wang, Y. Hypergraph transformer for semi-supervised classification. In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2024), Seoul, Republic of Korea, 14–19 April 2024; pp. 7515–7519.
23. Xia, L.H.; Huang, C.; Zhang, C.X. Self-supervised hypergraph transformer for recommender systems. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022), Washington, DC, USA, 14–18 August 2022; pp. 2100–2109.
24. Veličković, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep Graph Infomax. In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019.

25. Wu, H.; Li, N.; Zhang, J.; Chen, S.; Ng, M.K.; Long, J. Collaborative contrastive learning for hypergraph node classification. *Pattern Recognit.* **2024**, *146*, 109995.
26. Wei, T.X.; You, Y.N.; Chen, T.L.; Shen, Y.; He, J.; Wang, Z. Augmentations in hypergraph contrastive learning: Fabricated and generative. In Proceedings of the Thirty-Sixth Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022; pp. 1909–1922.
27. Cai, D.R.; Song, M.X.; Sun, C.X.; Zhang, B.; Hong, S.; Li, H. Hypergraph structure learning for hypergraph neural networks. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI 2022), Vienna, Austria, 23–29 July 2022; pp. 1923–1929.
28. Wang, Q.Q.; Zhao, B.W.; Ding, Z.M.; Zhang, X.; Gao, Q. A simple yet effective hypergraph clustering network. In Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2025), Montreal, QC, Canada, 16–22 August 2025; pp. 6352–6360.
29. Lee, D.J.; Shin, K. I’m Me, We’re Us, and I’m Us: Tri-directional contrastive learning on hypergraphs. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023), Washington, DC, USA, 7–14 February 2023; pp. 8456–8464.
30. Song, Y.M.; Gu, Y.; Li, T.Y.; Qi, J.; Liu, Z.; Jensen, C.S.; Yu, G. CHGNN: A semi-supervised contrastive hypergraph learning network. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 4515–4530. [CrossRef]
31. Qian, Y.Y.; Ma, T.Y.; Zhang, C.X.; Ye, Y. Dual-level hypergraph contrastive learning with adaptive temperature enhancement. In Proceedings of the Companion Proceedings of the ACM Web Conference 2024 (WWW 2024 Companion), Singapore, 13–17 May 2024; pp. 859–862.
32. Zhu, B.T.; Wang, X.B. Accurate trust evaluation for effective operation of social IoT systems via hypergraph-enabled self-supervised contrastive learning. In Proceedings of the 2025 IEEE International Conference on Communications (ICC 2025), Montreal, QC, Canada, 8–12 June 2025; pp. 1–6.
33. Li, Y.C.; Chen, H.X.; Sun, X.G.; Sun, Z.; Li, L.; Cui, L.; Yu, P.S.; Xu, G. Hyperbolic hypergraphs for sequential recommendation. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM 2021), Virtual, 1–5 November 2021; pp. 988–997.
34. Xie, H.Y.; Song, C.J.; Shao, H.; Wang, L. Hypergraph semi-supervised contrastive learning for hyperedge prediction based on enhanced attention aggregator. *Entropy* **2025**, *27*, 1046. [CrossRef] [PubMed]
35. Xia, X.; Yin, H.Z.; Yu, J.L.; Wang, Q.; Cui, L.; Zhang, X. Self-supervised hypergraph convolutional networks for session-based recommendation. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021), Virtual, 2–9 February 2021; pp. 4503–4511.
36. Liu, L.T.; Huang, F.; Liu, X.; Xiong, Z.; Li, M.; Song, C.; Zhang, W. Multi-view contrastive learning hypergraph neural network for drug-microbe-disease association prediction. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI 2023), Macao, China, 19–25 August 2023; pp. 4829–4837.

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.

Article

Optimized Deep Learning Framework for Emotion Recognition Using Multimodal Physiological Signals and Temporal Convolutional Networks

Mohsen Golafrouz ^{1,*}, Houshyar Asadi ^{1,*}, Mohammad Reza Chalak Qazani ², Anwar Hosen ¹, Zoran Najdovski ¹, Lei Wei ¹, Sam Oladazimi ¹ and Saeid Nahavandi ³

¹ Institute for Intelligent Systems Research and Innovation (IISRI), Geelong Waurin Ponds Campus, Deakin University, Waurin Ponds, VIC 3216, Australia

² School of Engineering, James Cook University, Townsville, QLD 4811, Australia

³ Associate Deputy Vice-Chancellor Research and Chief of Defence Innovation, Swinburne University of Technology, Hawthorn, VIC 3122, Australia

* Correspondence: s221323418@deakin.edu.au (M.G.); houshyar.asadi@deakin.edu.au (H.A.); Tel.: +61-415-601-288 (M.G.)

Abstract

Emotion recognition plays a crucial role in human–computer interaction, health monitoring, and affective computing by analysing physiological signals. Despite recent advancements, current research still faces challenges, including the lack of effective fusion strategies for diverse physiological modalities, difficulties in handling high-dimensional feature representations, and limited use of efficient temporal modelling techniques to capture complex emotional patterns. This study proposes a deep learning-based approach that fuses multiple physiological modalities, including Electroencephalography (EEG), Electrooculography (EOG), Electromyography (EMG), Galvanic Skin Response (GSR), Respiratory Rate (RR), Skin Temperature (SKT), and Photoplethysmography (PPG), to improve emotion recognition. Arousal and valence ratings were binarized into two classes (low/high) using a threshold of 4.5, formulating a binary classification problem. In addition to utilising Bidirectional Long Short-Term Memory (Bi-LSTM), the study employs Temporal Convolutional Networks (TCN), a widely used approach for time-series analysis, to efficiently capture temporal dependencies. The proposed model optimises feature selection through channel-wise strategies, incorporates advanced learning rate scheduling, and reduces computational overhead. Furthermore, window-wise, block-wise, and trial-wise evaluation protocols were investigated to assess the impact of temporal information leakage on emotion recognition performance. Using the DEAP dataset for validation, the proposed TCN-based approach achieved classification accuracies of 88.42% for valence and 86.35% for arousal under an overlapping block-wise evaluation protocol, demonstrating improved performance in binary emotion recognition and highlighting the importance of leakage-aware model assessment.

Keywords: emotion recognition; physiological signals; deep learning; multimodal fusion; temporal convolutional networks

1. Introduction

Emotional states profoundly impact how individuals perceive, process, and respond to information, affecting both cognition and psychological well-being. While positive

emotions foster psychological resilience, negative emotions are often linked to stress-related health issues [1,2]. Accurate emotion recognition is therefore essential across a wide range of applications, including mental health monitoring, adaptive human–computer interaction, and affective computing [3]. However, traditional emotion recognition methods relying on subjective assessments and manual annotations are time-consuming and prone to bias [4].

Recent advancements in artificial intelligence (AI), particularly deep learning, have significantly improved automated emotion recognition by leveraging physiological signals such as Electroencephalography (EEG), Electrooculography (EOG), Electromyography (EMG), Skin Temperature (SKT), Photoplethysmography (PPG), Galvanic Skin Response (GSR), and respiratory rate (RR) [5]. Although previous studies have explored deep learning with multimodal physiological signals, most have only utilised a subset of these modalities rather than integrating them comprehensively. Despite these improvements, challenges remain in feature selection, computational efficiency, and model scalability for real-time applications. Moreover, integrating deep learning with multimodal physiological signals is still evolving, constrained by limited datasets and model optimisation strategies.

Multimodal emotion recognition has gained increasing attention as it enables the integration of complementary information from multiple physiological signals to improve robustness and classification performance. Different modalities capture distinct aspects of emotional processing: EEG reflects neural activity, PPG captures cardiovascular responses, EOG records eye movement patterns, EMG measures muscle activity related to facial expressions, SKT reflects temperature changes associated with emotional states, and GSR represents skin conductance variations linked to arousal and stress. By combining these heterogeneous signals, multimodal approaches can better address the subjectivity and variability inherent in emotional responses, leading to more reliable emotion prediction systems [6].

To address these challenges, this study proposes a multimodal fusion framework for emotion recognition, incorporating EEG, PPG, GSR, EOG, EMG, SKT, and RR. Using the DEAP dataset, we preprocess signals, extract relevant features, and employ modality-specific subnetworks before integrating them into a unified deep learning model.

While multimodal deep learning approaches have been widely explored, this study investigates the effectiveness of Temporal Convolutional Networks (TCN) [7] and Bidirectional Long Short-Term Memory (Bi-LSTM) [8] for multimodal physiological emotion recognition. In addition to comparing temporal modelling capabilities, the study systematically examines the influence of different evaluation protocols, including window-wise, block-wise, and trial-wise validation strategies, to assess the impact of temporal information leakage on reported performance. Optimised feature selection and advanced learning rate scheduling techniques are further employed to improve accuracy while reducing computational overhead.

By exploring deep learning-based multimodal fusion strategies, this study aims to enhance the efficiency and scalability of emotion recognition systems, contributing to the development of robust, real-time applications in affective computing and related fields.

The main contributions of this study are summarised as follows:

1. A multimodal deep learning framework is proposed for emotion recognition by integrating seven physiological modalities, including EEG, PPG, GSR, EOG, EMG, SKT, and RR, enabling comprehensive exploitation of both central and peripheral physiological responses.
2. Multiple data partitioning strategies, including conventional window-wise, block-wise, and trial-wise evaluation protocols, are systematically investigated to examine the impact of temporal information leakage on emotion recognition performance.

3. A leakage-aware evaluation framework is introduced through the systematic investigation of window-wise, block-wise, non-overlapping block-wise, and trial-wise partitioning strategies, providing a more realistic assessment of model generalisation and quantifying the impact of temporal information leakage on reported emotion recognition performance.
4. A Temporal Convolutional Network (TCN)-based architecture is developed and evaluated against a Bi-LSTM baseline to investigate the effectiveness of temporal feature learning from multimodal physiological signals.
5. A modality-wise ablation study is conducted to quantify the contribution of individual physiological signals and to evaluate the effectiveness of multimodal fusion for emotion recognition.

Extensive experiments on the DEAP dataset demonstrate that the proposed framework achieves competitive performance while maintaining computational efficiency, supporting its potential applicability in real-time affective computing systems.

The remainder of this paper is organised as follows. Section 2 presents the background and reviews related work in multimodal emotion recognition. Section 3 describes the proposed methodology, including data preprocessing, feature extraction, model architecture, and evaluation protocols. Section 4 presents experimental results, including performance analysis, modality-wise ablation studies, and the investigation of temporal information leakage under different validation strategies. Finally, Section 5 concludes the paper and outlines future research directions.

2. Background

In recent years, numerous studies have explored emotion recognition using physiological signals and machine learning techniques. Verma and Tiwary [7] developed a multimodal fusion framework combining Discrete Wavelet Transform with classifiers such as Support Vector Machine (SVM), Multilayer Perceptron (MLP), and K-Nearest Neighbours (KNN) to classify emotions from EEG, GSR, and skin temperature. Yin and Zhao [8] introduced a multiple-fusion-layer-based ensemble classifier using stacked autoencoders (MESAE) to filter noise and extract stable features for arousal and valence classification. Khalili et al. [9] examined quadratic discriminant classifiers on EEG and peripheral signals, highlighting EEG's effectiveness, particularly when nonlinear features were employed. Chen et al. [10] proposed a multimodal fusion approach using EEG and peripheral signals, achieving optimal performance for arousal classification. García [11] applied Gaussian Process Latent Variable Models (GP-LVM) to map high-dimensional physiological data into a low-dimensional latent space, improving classification accuracy. Wagner et al. [12] investigated emotion recognition from music-induced stimuli using EMG, ECG, and skin conductivity, achieving high recognition rates with machine learning classifiers.

A notable study [13] introduced the MED4 dataset, recording EEG, PPG, speech, and facial images under controlled conditions. It demonstrated that EEG outperforms speech in emotion recognition and that fusing EEG with speech enhances robustness in noisy environments. The study also leveraged Temporal Convolutional Networks (TCNs) to improve temporal feature extraction and classification performance. Similarly, one study [14] demonstrated the effectiveness of TCNs for learning long-term temporal features from physiological signals, overcoming RNN limitations. Integrating the Connectionist Temporal Classification (CTC) algorithm further improved alignment of temporal features with sentiment labels. The results on the AMIGOS dataset showed that the TCN-based approach outperformed traditional methods, highlighting its potential for emotion recognition.

More recent research has increasingly shifted toward deep learning approaches for EEG-based emotion recognition. Bi-LSTM networks have demonstrated effectiveness in

modelling temporal dependencies in physiological signals [15], while CNN-RNN hybrid frameworks have improved spatial–temporal feature extraction [16]. Other studies have explored graph convolutional networks and transformer-based architectures to better capture inter-channel and temporal relationships in EEG signals [17]. The DEAP dataset [18] has become one of the most widely used benchmarks for evaluating multimodal physiological emotion recognition models.

Recent studies on the DEAP dataset have investigated multimodal fusion and advanced temporal modelling strategies for emotion recognition. Ensemble-based learning frameworks combining EEG, EMG, and EOG signals have demonstrated the effectiveness of heterogeneous physiological fusion for affect classification [19]. Deep canonical correlation analysis has also been explored to learn shared feature representations across modalities for arousal and valence recognition [20]. In addition, attention-based transformer architectures and CNN–LSTM hybrid models have shown strong performance in spatial–temporal representation learning and temporal decoding of EEG signals [21–23]. Transfer-learning approaches have been further investigated to improve robustness across subjects and modalities [24], while multimodal human–computer interaction studies integrating EEG with facial video have highlighted the complementary relationship between behavioural and physiological affective cues [25]. Together, these studies demonstrate growing interest in scalable and generalised deep learning frameworks for multimodal affective computing.

An important methodological consideration in physiological emotion recognition is the temporal segmentation of continuous signals. Many studies divide recordings into short overlapping windows to capture dynamic emotional variations and increase the number of training samples [26–28]. However, such overlapping segmentation can introduce information leakage when windows from the same trial are distributed across training and validation sets, potentially leading to inflated performance estimates [29]. To address this issue, more rigorous evaluation strategies, such as trial-wise and block-wise partitioning, have been proposed [30,31]. In particular, block-wise segmentation groups temporally adjacent windows into the same dataset partition, reducing the likelihood that highly correlated samples cross the train–validation boundary. This preserves the independence between training and validation data and provides a more reliable assessment of model generalisation.

Recent studies have increasingly emphasised the importance of evaluation methodology in EEG-based emotion recognition. A comprehensive review by Ma et al. [32] summarised numerous subject-independent deep learning studies and demonstrated that cross-subject evaluation on the DEAP dataset remains considerably more challenging than conventional within-subject settings due to substantial inter-subject variability. Brookshire et al. [29] discussed how data leakage can artificially inflate EEG deep learning performance and reduce the reliability of reported results. Lei et al. [31] investigated the impact of trial-wise splitting and temporal leakage in EEG-based emotion classification, showing that improper partitioning strategies may lead to overly optimistic accuracies. Kukhilava et al. [30] proposed a unified evaluation framework for EEG emotion recognition and highlighted inconsistencies in validation methodologies across existing studies. Similarly, Del Pup et al. [33] demonstrated that data partitioning strategies significantly influence EEG deep learning performance, while Liu et al. [34] introduced a comprehensive benchmark framework emphasising standardised and reproducible evaluation procedures.

In this study, a block-wise evaluation strategy is adopted to reduce temporal overlap between training and validation samples. Although this approach produces more conservative performance compared to conventional window-wise splitting, it provides a more realistic and unbiased assessment of model generalisation capability.

In addition to methodological studies on data leakage and evaluation protocols, several recent subject-independent EEG emotion recognition approaches have further demonstrated the challenges of cross-subject generalisation on the DEAP dataset. Quan et al. [35] proposed a deep learning framework for subject-independent EEG emotion recognition and reported accuracies of 79.59% for arousal and 81.19% for valence, highlighting the difficulty of maintaining robust performance under cross-subject evaluation settings. He et al. [36] introduced a domain adaptation-based approach for EEG emotion recognition, achieving 63.25% arousal and 64.33% valence accuracy, while emphasising the importance of reducing domain discrepancies between subjects. Together, these studies further confirm that subject-independent EEG emotion recognition remains substantially more challenging than conventional within-subject evaluation and reinforce the importance of rigorous and leakage-aware validation strategies.

Together, these studies indicate that DEAP continues to serve as a central benchmark for evaluating modality integration, feature representation, and temporal modelling strategies in affective computing. However, differences in modality selection, binary versus multi-class formulations, and cross-subject versus within-subject evaluation protocols make direct comparison challenging and highlight the need for reproducible multimodal pipelines with clearly defined learning objectives. Despite these advances, challenges such as computational complexity, overfitting, and limited real-time scalability persist. In particular, the integration of TCNs into multimodal frameworks remains underexplored, even though their ability to model long-term dependencies and enable parallelization offers clear benefits.

3. Methodology

3.1. Data Collection

The DEAP dataset [18], a widely used benchmark for emotion recognition, consists of multimodal physiological recordings from 32 participants (16 males, 16 females) aged between 19 and 37 years. The study was conducted in a controlled laboratory environment where participants watched 40 one-minute-long music video clips selected to elicit a range of emotional responses. These videos were chosen based on prior affective ratings to ensure diversity in emotional stimuli. During the experiment, physiological signals—including EEG, EOG, EMG, GSR, RR, SKT, and PPG—were recorded at a sampling rate of 128 Hz. EEG signals were captured from 32 channels following the international 10–20 system. After watching each video, participants self-reported their emotional states using a 9-point scale for arousal and valence. For this study, arousal and valence ratings were binarized, formulating a two-class classification problem. These subjective ratings serve as the ground truth labels for emotion classification.

For model training, the DEAP data were loaded in the pre-processed Python format, where each participant's file contains 40 trials. Each trial is represented as a three-dimensional array with the shape (40 trials $\times$ 40 channels $\times$ 8064 samples), corresponding to 32 EEG and 8 peripheral physiological signals. In this study, 11 EEG channels were selected based on prior literature and combined with six peripheral physiological channels (EOG, EMG, GSR, RR, SKT, and PPG), resulting in a final configuration of 17 channels for analysis. This multimodal configuration enables a comprehensive assessment of both central nervous system activity (EEG) and peripheral autonomic physiological responses. The corresponding label array has the shape (40 $\times$ 4), representing valence, arousal, dominance, and liking, where only valence and arousal dimensions were used for classification.

3.2. Preprocessing and Feature Extraction

To prepare the data for deep learning, each physiological signal underwent a multi-stage preprocessing and segmentation pipeline designed to enhance signal quality and reduce redundancy. First, baseline noise and artefacts were removed using the DEAP dataset's preprocessing procedures, including filtering and normalisation. From the 40 available channels, 11 EEG channels were selected and combined with six peripheral physiological signals (EOG, EMG, GSR, RR, SKT, and PPG), resulting in 17 channels used for subsequent analysis.

Each trial (8064 samples) was segmented into overlapping windows of 2 s (256 samples) with a 0.125 s step (16 samples). For each window, a Fast Fourier Transform (FFT) was applied to extract spectral band-power features across five frequency bands commonly used in emotion-related EEG and physiological analysis. Band-power features from all selected channels were concatenated into a single feature vector per window.

To prevent overestimation of model performance caused by the high overlap between consecutive windows, a block-wise splitting strategy was employed. The sequence of overlapping windows from each trial was partitioned into non-overlapping contiguous blocks, each containing 8 consecutive windows. Given the windowing configuration, these blocks represent short, highly correlated temporal segments of the original signal rather than independent samples. Blocks were constructed sequentially without shuffling to preserve temporal dependencies. A deterministic partitioning scheme was applied in which every fourth block was assigned to the validation set, while the remaining blocks were used for training.

To mitigate the effects of temporal information leakage caused by highly overlapping windows, a block-wise splitting strategy was initially employed. The sequence of overlapping windows from each trial was partitioned into contiguous blocks, each containing eight consecutive windows, and a deterministic partitioning scheme was applied in which every fourth block was assigned to the validation set. This strategy reduces the likelihood of highly similar windows appearing in both training and validation sets; however, some temporal correlation may still remain due to the substantial overlap between neighbouring windows. To further investigate the influence of evaluation methodology on reported performance, additional experiments using larger step sizes, non-overlapping block-wise partitioning, and strict trial-wise validation were conducted. These evaluation protocols are described in Section 4.6.

All extracted features and corresponding labels were stored as NumPy arrays to enable efficient data loading during model training. Training and validation were performed in a within-subject manner using block-wise configuration to provide a fair, reproducible, and more rigorous assessment while reducing temporal information leakage between training and validation samples.

Evaluation Protocols

Several evaluation protocols were investigated to assess the impact of temporal information leakage on emotion recognition performance. First, a conventional window-wise strategy was considered, in which overlapping windows were randomly partitioned between training and validation sets. Second, a block-wise strategy grouped neighbouring windows into contiguous temporal blocks before partitioning. Third, a leakage-reduced block-wise protocol was implemented by increasing the window step size, thereby reducing overlap between adjacent windows. Fourth, a non-overlapping block-wise protocol was adopted using a step size equal to the window length. Finally, a strict trial-wise protocol was employed in which entire trials were assigned exclusively to either the training or

validation set. These protocols enabled a systematic investigation of how temporal overlap influences reported classification performance.

3.3. Temporal Convolutional Networks (TCN) and Bi-LSTM for Time-Series Modelling

Temporal Convolutional Networks (TCNs) [37] have recently gained significant attention as a powerful alternative to recurrent neural networks (RNNs) for time-series and sequential data modelling. Unlike traditional recurrent architectures such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) [38], which rely on sequential processing of data over time steps, TCNs leverage the strengths of convolutional operations to capture dependencies across long temporal spans in a more efficient manner. Specifically, TCNs employ causal convolutions, which ensure that the prediction at any time step is based only on the current and previous inputs, thereby preserving temporal order and avoiding information leakage from the future.

Another important feature of TCNs is the use of dilated convolutions, which systematically increase the receptive field of the convolutional filters without requiring a proportional increase in network depth. Through this mechanism, TCNs can capture both short- and long-range dependencies while maintaining computational efficiency. This design contrasts with recurrent approaches, where modelling long-term dependencies is often hindered by vanishing or exploding gradients. By expanding the receptive field exponentially with each layer, TCNs can learn temporal patterns over extended time spans without recurrent connections. Moreover, the convolutional nature of TCNs enables parallel processing of entire sequences, which substantially improves training speed compared to sequential models like Bi-LSTM.

In this study, we employ TCNs as the primary model for emotion recognition using multimodal physiological signals, including EEG, PPG, GSR, EOG, EMG, RR, and SKT. These signals inherently contain complex temporal dynamics that require models capable of efficiently extracting both local and global dependencies. The detailed architecture of the proposed TCN model is illustrated in Figure 1.

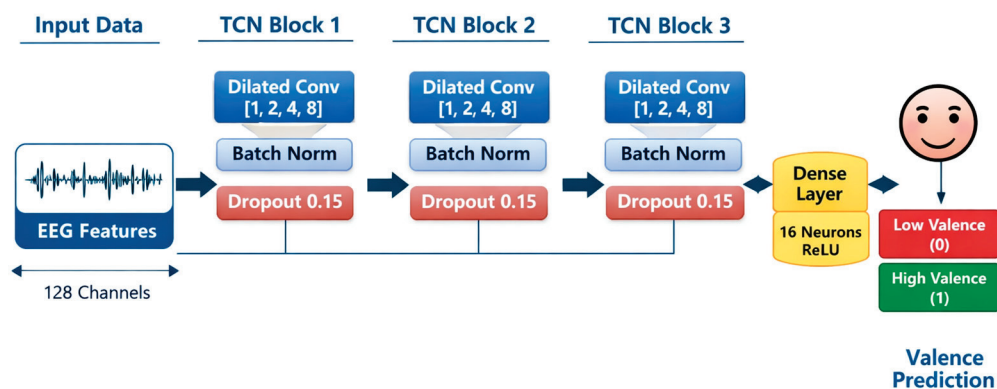


Figure 1. Architecture of the proposed TCN model. The figure was created using ChatGPT (OpenAI, GPT-4) and subsequently refined by the authors.

For benchmarking and comparison, we also implement a Bi-LSTM model, as it is one of the most widely used architectures in time-series analysis due to its ability to capture bidirectional contextual information. However, despite its effectiveness, Bi-LSTM suffers from limitations such as higher computational cost, slower training, and reduced scalability for real-time applications. In addition, recurrent architectures often struggle to capture very long-range dependencies efficiently, a challenge that TCNs address through their hierarchical convolutional design.

By conducting a comparative analysis between TCN and Bi-LSTM, we aim to highlight the advantages of TCNs in terms of both predictive performance and computational

efficiency. The goal of this work is to demonstrate the suitability of TCNs for physiological signal-based emotion recognition, emphasising their ability to extract meaningful temporal representations while maintaining stable training dynamics. This evaluation not only validates the robustness of TCNs but also provides insights into their practical applicability for large-scale and real-time affective computing systems.

3.4. Model Architecture

In contrast to recurrent models such as Bi-LSTM, this study adopts a TCN-based architecture for modelling physiological signals. The TCN structure is specifically designed to handle long-range temporal dependencies while offering superior parallelization and stability during training. The architecture is composed of multiple temporal convolutional layers, which progressively extract temporal features from the input signals. Each convolutional layer is embedded within residual blocks, ensuring that information can flow across layers without degradation and improving the model's ability to learn deep temporal representations.

Feature vectors from each modality were concatenated along the feature dimension prior to input into the shared TCN backbone, forming a unified multimodal representation. To enhance generalisation and prevent overfitting, the architecture incorporates dropout layers and batch normalisation after convolutional operations. Dropout helps reduce reliance on specific filters, while batch normalisation stabilises learning and accelerates convergence. At the classification stage, fully connected layers are employed to map the extracted temporal features into two discrete classes for arousal and valence, enabling effective binary emotion recognition.

The proposed TCN model was implemented using the TensorFlow/Keras deep-learning framework. The network consisted of three stacked TCN layers with filter sizes of 64, 128, and 32, respectively. Each TCN layer employed a kernel size of 3 and dilation factors of [1, 2, 4, 8] to capture both short- and long-term temporal dependencies. Batch normalisation and dropout (rate = 0.15) were applied after each TCN layer to improve training stability and reduce overfitting. For classification, a fully connected dense layer with 16 neurons and ReLU activation was used, followed by a single-neuron output layer with a sigmoid activation function for binary classification. The main architectural parameters are summarised in Table 1.

Table 1. TCN model architecture and hyperparameters.

Layer	Kernel Size	Dilations	Filters	Dropout/BN/Activation
TCN Block 1	3	[1, 2, 4, 8]	64	BatchNorm + Dropout 0.15
TCN Block 2	3	[1, 2, 4, 8]	128	BatchNorm + Dropout 0.15
TCN Block 3	3	[1, 2, 4, 8]	32	BatchNorm + Dropout 0.15
Dense	N/A	N/A	16	ReLU
Output	N/A	N/A	1	Sigmoid

Training was performed using the Adam optimizer (learning rate = 0.0001) with a ReduceLROnPlateau scheduler that halved the learning rate when the validation loss plateaued (patience = 5). Binary cross-entropy loss was used, and the model was trained for 230 epochs with a batch size of 256.

A block-wise train-validation split was initially adopted for model development. Subsequently, additional leakage-aware evaluation protocols, including non-overlapping block-wise and trial-wise validation, were investigated to assess the impact of temporal in-

formation leakage on classification performance. All experiments followed a within-subject training protocol. Separate networks were trained for arousal and valence prediction.

Overall, the TCN architecture offers several advantages over recurrent counterparts. It enables faster training by exploiting convolutional operations, ensures stable gradient flow through residual connections, and scales effectively to long sequences due to dilated convolutions. These properties make the proposed TCN model well-suited for real-time physiological signal-based emotion recognition, where both computational efficiency and accurate temporal modelling are critical. The architecture is illustrated in Figure 2.



Figure 2. Architecture of the proposed TCN Model.

3.4.1. Bidirectional LSTM Model for Comparison

To provide a comparative baseline for the proposed TCN architecture, a Bidirectional Long Short-Term Memory (Bi-LSTM) model was implemented to capture sequential dependencies in both forward and backward directions. The Bi-LSTM is particularly effective in modelling long-term temporal relationships, allowing the network to retain past and future contextual information in time-series data.

The model consisted of three LSTM layers with 64, 128, and 32 hidden units, respectively. All layers were implemented as bidirectional LSTMs to enhance temporal feature extraction from both forward and backward directions. Dropout layers (0.15, 0.15, and 0.15) were applied after each LSTM block, and Batch Normalisation was used to stabilise and accelerate convergence. The network concluded with a fully connected dense layer of 16 ReLU-activated units and a sigmoid output layer for binary classification.

Training was performed using the Adam optimizer with an initial learning rate of 0.0001. Binary cross-entropy loss was used for optimisation. The model was trained for 230 epochs with a batch size of 256. To prevent temporal leakage caused by highly overlapping windows, a block-wise train-validation splitting strategy was adopted, where consecutive windows from the same trial were grouped and assigned entirely to either the training or validation set. All experiments were implemented in Python using the TensorFlow deep learning framework under a within-subject evaluation protocol consistent with the proposed TCN model.

3.4.2. Ablation Study for Modality Contribution

To assess the contribution of individual physiological modalities, an ablation study was conducted using the same TCN architecture. Models were trained on subsets of modalities, including EEG only, EEG with EOG and EMG, EEG with EOG, EMG, and SKT, and all modalities combined. This analysis provides quantitative insight into the incremental benefit of including peripheral signals alongside EEG and supports the evaluation of the proposed early-fusion approach.

3.5. Adaptive Learning Rate Adjustment

Learning rate adjustment is essential for balancing the speed and stability of model convergence. If the learning rate is too small, training progresses slowly, while a high learning rate can cause instability and prevent the model from reaching an optimal solution. Adaptive adjustment [39] methods dynamically modify the learning rate based on

validation metrics, such as loss or accuracy, rather than following predefined schedules. When performance plateaus, these methods reduce the learning rate, allowing for smaller, more precise updates. This helps the model escape stagnation and continue improving, especially in later training stages where large updates could hinder fine-tuning.

4. Results and Discussion

4.1. Model Performance Overview

Unless otherwise specified, the detailed analyses presented in this section correspond to the valence classification task. For detailed analysis of model behaviour, including convergence dynamics and computational efficiency, the valence task is used as a representative case. This avoids redundancy in reporting, as both arousal and valence models share identical architectures and training protocols but require separate training processes. Arousal classification results are reported separately in Section 4.7 and summarised under the same experimental conditions.

To provide a more rigorous evaluation, a block-wise data partitioning strategy was initially adopted, where temporally adjacent overlapping windows were grouped into contiguous blocks and assigned exclusively to either the training or validation set. This approach reduces the risk of temporal information leakage caused by highly overlapping windows and provides a more conservative assessment of model generalisation compared with conventional window-wise validation.

4.2. Accuracy and Convergence Analysis

The performance of Temporal Convolutional Networks (TCN) was evaluated against a Bi-LSTM model, summarised in Table 2. The table presents key efficiency metrics such as training accuracy, validation accuracy, average training time per epoch, and inference latency per sample. Inference latency was measured per 5 s input window.

Table 2. Comparison of TCN and Bi-LSTM performance for valence classification under overlapping block-wise evaluation (window size = 256, step size = 16, block size = 8).

Model	Trainable Parameters	Train Acc	Validation Acc	Avg. Train Time/Epoch	Inference Latency per 5 s Window
TCN	187,681	97.26%	88.42%	14.12 s	0.32 ms
Bi-LSTM	373,793	84.93%	78.87%	15.32 s	0.42 ms

TCN consistently outperforms Bi-LSTM in both training and validation accuracy. It achieves a training accuracy of 97.26% and a validation accuracy of 88.42%, outperforming Bi-LSTM by approximately 9–10%. These results indicate that TCN's causal and dilated convolutions are more effective in capturing both short- and long-term temporal dependencies in physiological signals. The small gap between training and validation accuracy also suggests that TCN generalises better and mitigates overfitting compared to Bi-LSTM.

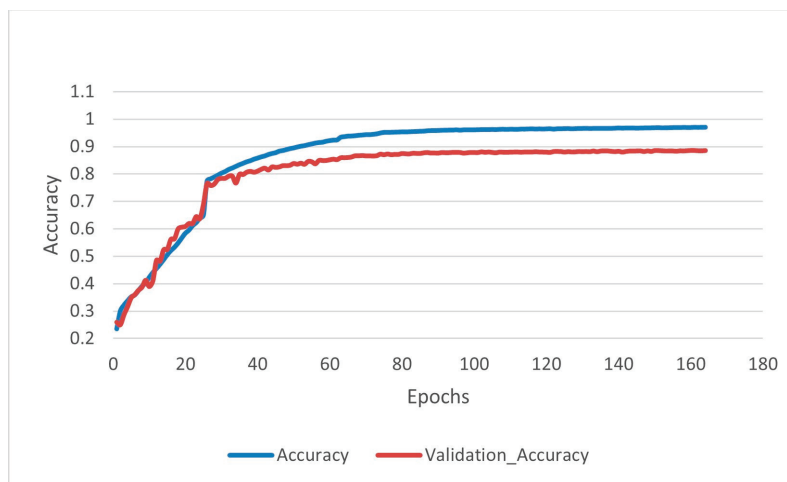
It is worth noting that when conventional window-wise splitting is applied—where individual overlapping windows are randomly partitioned—the TCN model achieves training and validation accuracy exceeding 90%. However, such splitting may allow temporally overlapping segments from the same trial to appear in both training and validation sets, potentially inflating performance estimates due to shared signal content. By contrast, the adopted block-wise partitioning strategy groups adjacent windows into contiguous segments before splitting, reducing the likelihood that highly correlated samples appear in both training and validation sets. This results in a more conservative yet more realistic evaluation of classification performance and model generalisation.

4.3. Computational Efficiency

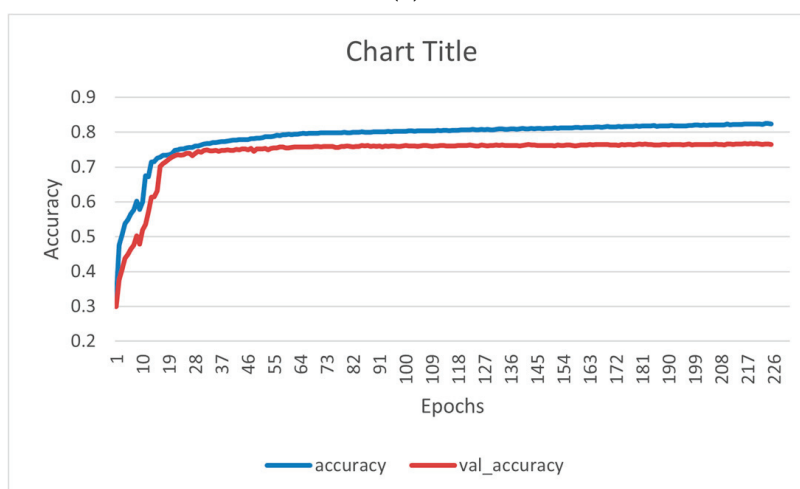
The average training time per epoch demonstrates computational efficiency: TCN required approximately 14.12 s per epoch, slightly faster than Bi-LSTM at 15.32 s. This improvement reflects the parallelized convolutional operations in TCN, which efficiently process sequential data. Additionally, inference latency is significantly lower for TCN (0.32 ms/sample) than Bi-LSTM (0.42 ms/sample), highlighting TCN's suitability for real-time emotion recognition systems.

4.4. Learning Dynamics

The accuracy vs. epochs plot (Figure 3) illustrates the learning progression of the models over training iterations. The TCN model (top) exhibits a steady and rapid increase in accuracy, reaching a plateau at higher epochs, while the Bi-LSTM and LSTM models (bottom) show a comparatively slower convergence. The smoother and more stable accuracy curve of TCN indicates improved learning efficiency and reduced fluctuations, reflecting its better capacity to capture temporal dependencies in physiological signals. The gap between training and validation accuracy remains minimal, further confirming stable generalisation under the stricter block-wise evaluation setting.



(a)



(b)

Figure 3. Accuracy vs. Epochs for (a) TCN (top) and (b) Bi-LSTM (bottom) for valence classification.

Dynamic learning rate adjustment strategies played a crucial role in enhancing model convergence and overall performance. Adaptive scheduling effectively fine-tuned network parameters, preventing overshooting and improving stability during training. As seen in the accuracy vs. epochs plot, the learning curve exhibits distinct step-like increases in accuracy, which align with scheduled reductions in the learning rate. These jumps indicate moments when the optimizer adjusts step sizes, allowing the model to escape local minima and refine its weight updates more effectively. Such adaptive techniques helped mitigate overfitting by ensuring a balanced optimisation process. The results emphasise the importance of learning rate management in training deep neural networks for physiological signal-based emotion recognition.

4.5. Ablation Study Results

The ablation study (Table 3) was conducted to evaluate the contribution of each physiological modality to the performance of the proposed TCN model. As shown, the model trained using EEG only achieves a baseline validation accuracy of 80.5%, indicating that central nervous system activity alone provides meaningful but limited discriminative information for emotion recognition.

Table 3. Validation accuracies for different modality combinations for valence classification.

Modalities	Train Acc (%)	Validation Acc (%)
EEG only	85.6%	80.5%
EEG + EOG + EMG	88.2%	81.8%
EEG + EOG + EMG + SKT	93.3%	85.3%
All modalities	92.9%	88.42%

When peripheral modalities (EOG and EMG) are added, the validation accuracy increases to 81.8%, demonstrating that ocular and muscular signals contribute complementary temporal features that enhance classification capability. Further inclusion of SKT leads to a more substantial improvement, raising validation accuracy to 85.3%. This indicates that thermoregulatory responses provide additional discriminative information that is not captured by EEG and facial-related signals alone.

The full multimodal configuration achieves the highest validation accuracy (88.42%), confirming that combining central (EEG) and multiple peripheral physiological signals yields the most robust performance. Although the training accuracy slightly decreases compared to the previous configuration (92.9% vs. 93.3%), the validation accuracy improves, suggesting better generalisation and reduced overfitting.

Overall, the ablation results demonstrate that each additional modality contributes complementary information, validating the effectiveness of the early-fusion strategy adopted in the TCN framework and highlighting the importance of multimodal physiological integration for emotion recognition.

4.6. Impact of Evaluation Protocols and Temporal Leakage

To investigate the influence of temporal information leakage on emotion recognition performance, several evaluation strategies were examined, including conventional window-wise validation, leakage-reduced block-wise validation, non-overlapping block-wise validation, and strict trial-wise validation, summarised in Table 4.

Table 4. Impact of evaluation protocols on validation accuracy for valence classification.

Evaluation Protocol	Window Size	Step Size	Block Size	Train–Validation Leakage	Validation Accuracy (%)
Window-wise	256	16	-	High	93.67
Block-wise	256	16	8	High	88.42
Block-wise	256	128	8	Reduced	73.2
Non-overlapping Block-wise	256	256	8	Minimal	69.71
Trial-wise	256	16	-	None	68.13

The results indicate that performance decreases as stricter evaluation protocols are adopted. While conventional window-wise validation achieves the highest accuracy, it is susceptible to temporal information leakage caused by highly overlapping windows. Increasing the step size and employing block-wise partitioning reduces leakage and produces more conservative performance estimates. The trial-wise protocol provides the most rigorous evaluation by ensuring complete separation of trials between training and validation sets, thereby offering the most realistic assessment of model generalisation. Although trial-wise evaluation yields lower accuracy than window-wise and block-wise approaches, the resulting performance estimates are more reliable because they better reflect the model's ability to generalise to unseen trials. This observation is particularly important for the DEAP dataset, where the number of independent trials is limited, making leakage-aware evaluation essential for obtaining realistic and reproducible performance estimates.

4.7. Comparison with Previous Studies

Direct comparison between emotion recognition studies remains challenging because reported accuracies are strongly influenced by the adopted evaluation methodology, including segmentation strategy, overlap configuration, subject dependency, and data partitioning protocol. Recent reviews have highlighted that evaluation protocols can substantially affect reported performance and that studies employing different validation strategies should not be compared directly. In particular, conventional window-wise and random within-subject partitioning approaches may introduce temporal information leakage when highly overlapping windows are distributed across both training and validation sets, leading to overly optimistic performance estimates. Conversely, rigorous evaluation protocols, including non-overlapping, block-wise, trial-wise, cross-subject, cross-domain, and subject-independent approaches, provide more conservative but more reliable assessments of model generalisation.

To facilitate a fair comparison, previous studies are divided into two categories according to their evaluation methodology. Table 5 summarises studies that employed conventional evaluation protocols susceptible to temporal information leakage, whereas Table 6 presents studies adopting more rigorous evaluation and generalisation-oriented validation strategies. Following recent recommendations in the literature, studies employing fundamentally different evaluation protocols are reported separately to avoid misleading direct comparisons. Although all studies were evaluated using the DEAP dataset, differences in preprocessing pipelines, modality selection, feature extraction methods, and validation protocols should be considered when interpreting the reported accuracies.

Table 5. Comparison with previous studies using conventional evaluation protocols. Bold values indicate the best performance among the compared methods.

Ref	Modalities	Evaluation Protocol	Accuracy (%)
Wang et al. [22]	EEG	Subject-dependent	91.9–92.3 (V)
Alhagry et al. [17]	EEG, EMG, EOG	Within-subject	90.74 (V)
Qiu et al. [20]	EEG + Peripheral Signals	Within-subject	88.51 (V)
Proposed TCN	EEG, EOG, EMG, SKT, GSR, RR, PPG	Overlapping Block-wise (16,8)	88.42 (V), 86.35 (A)
Proposed TCN	EEG, EOG, EMG, SKT, GSR, RR, PPG	Window-wise	93.67 (V) 91.24 (A)

Table 6. Comparison with studies employing rigorous evaluation protocols. Bold values indicate the best performance among the compared methods.

Study	Modalities	Evaluation Protocol	Accuracy (%)
Lei et al. [31]	EEG	Trial-wise	67.0 ± 18.0 (V)
He et al. [36]	EEG	Cross-domain	64.33 (V)
Sartipi et al. [40]	EEG	Cross-subject	61.00 (V)
Proposed TCN	EEG, EOG, EMG, SKT, GSR, RR, PPG	Non-overlapping Block-wise	69.71 (V)
Proposed TCN	EEG, EOG, EMG, SKT, GSR, RR, PPG	Trial-wise	68.13 (V)

The studies presented in Table 5 generally report high recognition accuracies; however, their evaluation protocols may permit temporal information leakage between training and validation sets, particularly when highly overlapping windows are employed. This can result in overly optimistic estimates of model performance and may partially explain the exceptionally high accuracies reported by several studies.

Compared with the studies presented in Table 5, the accuracies reported in Table 6 are noticeably lower. However, these results provide a more realistic estimate of model generalisation because they are obtained under stricter evaluation protocols that reduce or eliminate temporal information leakage. The studies in Table 6 adopt rigorous validation methodologies, including trial-wise, cross-subject, cross-domain, and leakage-aware block-wise evaluation, all of which aim to provide more reliable performance estimates than conventional window-wise validation. The proposed framework achieved 68.13% accuracy under strict trial-wise validation and 69.71% under non-overlapping block-wise evaluation, demonstrating competitive performance compared with other rigorous evaluation approaches. Furthermore, the substantial performance difference between Tables 5 and 6 highlights the influence of evaluation methodology on reported emotion recognition accuracy. Although the trial-wise and non-overlapping block-wise accuracies are lower than those obtained using conventional window-wise validation, they offer a more reliable assessment of real-world performance. This observation is particularly important for the DEAP dataset, where the number of independent trials is limited, and the use of highly overlapping windows can otherwise lead to inflated performance estimates.

A further limitation of the present study is that the primary experiments were conducted under a within-subject evaluation framework. Although within-subject validation reduces inter-subject variability and enables a focused assessment of temporal modelling performance, it does not fully capture the challenges associated with generalising emotion

recognition models to unseen individuals. Physiological responses to emotional stimuli can vary substantially across participants, making cross-subject emotion recognition considerably more difficult. Consequently, future research should further investigate subject-independent and cross-subject validation strategies to evaluate the robustness and practical applicability of the proposed framework in real-world scenarios.

5. Conclusions

Recognising emotions through multiple physiological modalities simultaneously is crucial for enabling real-time emotion recognition and improving classification accuracy. While traditional approaches often rely on single-modality signals, limiting their robustness and generalisability, this study addresses key gaps in the field—namely, the lack of effective fusion strategies for diverse physiological inputs, the challenge of handling high-dimensional time-series data, and the underutilization of efficient temporal modelling techniques.

To address these challenges, we proposed a deep learning framework that integrates EEG, PPG, GSR, EOG, EMG, RR, and SKT using an early-fusion strategy combined with advanced temporal modelling. A TCN was employed as the primary architecture and benchmarked against Bi-LSTM to evaluate sequential modelling effectiveness.

A key contribution of this work is the investigation of leakage-aware evaluation strategies, including block-wise and trial-wise validation protocols, to reduce information leakage caused by highly overlapping windows and provide a more realistic assessment of model generalisation.

Experimental results on the DEAP dataset demonstrate that the proposed TCN model achieves a validation accuracy of 88.42% for valence and 86.35% for arousal under the stricter block-wise protocol, outperforming the Bi-LSTM baseline. In addition, the modality-wise ablation study confirms that integrating peripheral physiological signals alongside EEG progressively improves classification performance, with the full multimodal configuration achieving the highest validation accuracy. These findings highlight the complementary nature of central and peripheral signals and validate the effectiveness of the proposed fusion framework.

Furthermore, an additional investigation was conducted to examine the influence of evaluation protocols and temporal information leakage on emotion recognition performance. The results demonstrated that classification accuracy decreases as increasingly rigorous validation strategies are adopted, with trial-wise and non-overlapping block-wise protocols providing more conservative but more reliable estimates of model generalisation. These findings highlight the importance of leakage-aware evaluation methodologies and suggest that conventional window-wise validation may overestimate performance when highly overlapping windows are used.

While the proposed framework demonstrates strong performance, certain limitations should be acknowledged. The study relies on a relatively small sample size and employs a within-subject evaluation protocol, which may restrict generalisation to unseen participants. Future work will focus on expanding the dataset by collecting custom multimodal physiological recordings and exploring cross-subject validation to further enhance the robustness and applicability of the model in real-world scenarios.

Author Contributions: Conceptualization, M.G. and H.A.; methodology, M.G.; software, M.G.; validation, M.G., H.A. and M.R.C.Q.; formal analysis, M.G.; investigation, M.G.; resources, H.A., A.H., Z.N., L.W., S.O. and S.N.; data curation, M.G.; writing—original draft preparation, M.G.; writing—review and editing, H.A., M.R.C.Q., A.H., Z.N., L.W., S.O. and S.N.; visualization, M.G.; supervision, H.A. and M.R.C.Q.; project administration, H.A.; funding acquisition, H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Australian Research Council (ARC), Discovery Project, grant number DP260101427.

Data Availability Statement: The data used in this study are publicly available from the DEAP dataset repository: <https://www.eecs.qmul.ac.uk/mmv/datasets/deap/> (accessed on 8 June 2026). Processed data and source code are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the reviewers for their valuable comments and suggestions.

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

References

- Hag, A.; Al-Shargie, F.; Handayani, D.; Asadi, H. Mental Stress Classification Based on Selected Electroencephalography Channels Using Correlation Coefficient of Hjorth Parameters. *Brain Sci.* **2023**, *13*, 1340. [CrossRef]
- Shajari, A.; Asadi, H.; Glaser, S.; Arogbonlo, A.; Mohamed, S.; Kooijman, L.; Abu Alqumsan, A.; Nahavandi, S. Detection of driving distractions and their impacts. *J. Adv. Transp.* **2023**, *2023*, 2118553. [CrossRef]
- D'mello, S.K.; Kory, J. A review and meta-analysis of multimodal affect detection systems. *ACM Comput. Surv. (CSUR)* **2015**, *47*, 1–36. [CrossRef]
- Zeng, Z.; Pantic, M.; Roisman, G.I.; Huang, T.S. A survey of affect recognition methods: Audio, visual and spontaneous expressions. In *Proceedings of the 9th International Conference on Multimodal Interfaces*; IEEE: New York, NY, USA, 2007; pp. 126–133.
- Vairamani, A.D. Advancements in multimodal emotion recognition: Integrating facial expressions and physiological signals. In *Affective Computing for Social Good: Enhancing Well-Being, Empathy, and Equity*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 217–240.
- Zhao, S.; Jia, G.; Yang, J.; Ding, G.; Keutzer, K. Emotion recognition from multiple modalities: Fundamentals and methodologies. *IEEE Signal Process. Mag.* **2021**, *38*, 59–73. [CrossRef]
- Verma, G.K.; Tiwary, U.S. Multimodal fusion framework: A multiresolution approach for emotion classification and recognition from physiological signals. *NeuroImage* **2014**, *102*, 162–172. [CrossRef] [PubMed]
- Yin, Z.; Zhao, M.; Wang, Y.; Yang, J.; Zhang, J. Recognition of emotions using multimodal physiological signals and an ensemble deep learning model. *Comput. Methods Programs Biomed.* **2017**, *140*, 93–110. [CrossRef] [PubMed]
- Khalili, Z.; Moradi, M.H. Emotion recognition system using brain and peripheral signals: Using correlation dimension to improve the results of EEG. In *2009 International Joint Conference on Neural Networks*; IEEE: New York, NY, USA, 2009; pp. 1571–1575.
- Chen, J.; Hu, B.; Xu, L.; Moore, P.; Su, Y. Feature-level fusion of multimodal physiological signals for emotion recognition. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*; IEEE: New York, NY, USA, 2015; pp. 395–399.
- García, H.F.; Álvarez, M.A.; Orozco, Á.Á. Gaussian process dynamical models for multimodal affect recognition. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*; IEEE: New York, NY, USA, 2016; pp. 850–853.
- Wagner, J.; Kim, J.; André, E. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *2005 IEEE International Conference on Multimedia and Expo*; IEEE: New York, NY, USA, 2005; pp. 940–943.
- Wang, Q.; Wang, M.; Yang, Y.; Zhang, X. Multi-modal emotion recognition using EEG and speech signals. *Comput. Biol. Med.* **2022**, *149*, 105907. [CrossRef]
- Sheng, Y.; Hu, Q.; Cao, J. Physiological Signal Emotion Recognition Based on Temporal Convolutional Networks. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2022; Volume 2258, p. 012034.
- Zhu, M.; Wang, Q.; Luo, J. Emotion recognition based on dynamic energy features using a bi-LSTM network. *Front. Comput. Neurosci.* **2022**, *15*, 741086. [CrossRef] [PubMed]
- Li, X.; Song, D.; Zhang, P.; Zhang, Y.; Hou, Y.; Hu, B. Exploring EEG features in cross-subject emotion recognition. *Front. Neurosci.* **2018**, *12*, 162. [CrossRef]
- Alhagry, S.; Fahmy, A.A.; El-Khoribi, R.A. Emotion recognition based on EEG using LSTM recurrent neural network. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 358. [CrossRef]
- Koelstra, S.; Muhl, C.; Soleymani, M.; Lee, J.-S.; Yazdani, A.; Ebrahimi, T.; Pun, T.; Nijholt, A.; Patras, I. Deap: A database for emotion analysis; using physiological signals. *IEEE Trans. Affect. Comput.* **2011**, *3*, 18–31. [CrossRef]
- Zhang, J.; Zhang, Y.; Zhan, S.; Cheng, C. Ensemble emotion recognizing with multiple modal physiological signals. *arXiv* **2020**, arXiv:2001.00191.
- Qiu, J.-L.; Liu, W.; Lu, B.-L. Multi-view emotion recognition using deep canonical correlation analysis. In *International Conference on Neural Information Processing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 221–231.

21. Li, Q.; Liu, Y.; Yan, F.; Zhang, Q.; Liu, C. Emotion recognition based on multiple physiological signals. *Biomed. Signal Process. Control.* **2023**, *85*, 104989. [CrossRef]
22. Wang, T.; Huang, X.; Xiao, Z.; Cai, W.; Tai, Y. EEG emotion recognition based on differential entropy feature matrix through 2D-CNN-LSTM network. *EURASIP J. Adv. Signal Process.* **2024**, *2024*, 49. [CrossRef]
23. Xu, Y.; Du, Y.; Li, L.; Lai, H.; Zou, J.; Zhou, T.; Xiao, L.; Liu, L.; Ma, P. AMD-ET: Attention based multiple dimensions EEG transformer for emotion recognition. *IEEE Trans. Affect. Comput.* **2023**, *15*, 1067–1077. [CrossRef]
24. Fu, Z.; Zhang, B.; He, X.; Li, Y.; Wang, H.; Huang, J. Emotion recognition based on multi-modal physiological signals and transfer learning. *Front. Neurosci.* **2022**, *16*, 1000716. [CrossRef] [PubMed]
25. Hassan, C.A.U.; Ehatisham-ul-Haq, M.; Murtaza, F.; Yasin, A.U.; Ullah, S.S. EmoTrans attention based emotion recognition using EEG signals and facial analysis with expert validation. *Sci. Rep.* **2025**, *15*, 22004. [CrossRef]
26. Cao, L.; Wu, H.; Chen, S.; Dong, Y.; Zhu, C.; Jia, J.; Fan, C. A novel deep learning method based on an overlapping time window strategy for brain–computer interface-based stroke rehabilitation. *Brain Sci.* **2022**, *12*, 1502. [CrossRef]
27. Garg, S.; Patro, R.K.; Behera, S.; Tigga, N.P.; Pandey, R. An overlapping sliding window and combined features based emotion recognition system for EEG signals. *Appl. Comput. Inform.* **2025**, *21*, 114–130. [CrossRef]
28. Rehman, A.; Haldal, I.; Lin, J.C.-W. Spatiotemporal EEG-Based Emotion Recognition Using SAM Ratings from Serious Games with Hybrid Deep Learning. *arXiv* **2025**, arXiv:2508.21103.
29. Brookshire, G.; Kasper, J.; Blauch, N.M.; Wu, Y.C.; Glatt, R.; Merrill, D.A.; Gerrol, S.; Yoder, K.J.; Quirk, C.; Lucero, C.; et al. Data leakage in deep learning studies of translational EEG. *Front. Neurosci.* **2024**, *18*, 1373515. [CrossRef] [PubMed]
30. Kukhilava, N.; Tsmindashvili, T.; Kalandadze, R.; Gupta, A.; Katamadze, S.; Brémond, F.; Ferrari, L.M.; Müller, P.; Wirth, B.E. Evaluation in eeg emotion recognition: State-of-the-art review and unified framework. *arXiv* **2025**, arXiv:2505.18175.
31. Lei, P.; Wu, M.; Yi, W.; Mo, H. Impact of Trial-wise and Test Data Leakage on EEG-Based Emotion Classification. In Proceedings of the International IJCAI Workshop on 1st Challenge and Workshop for 4D Micro-Expression Recognition for Mind Reading, Guangzhou, China, 29 August 2025.
32. Ma, W.; Zheng, Y.; Li, T.; Li, Z.; Li, Y.; Wang, L. A comprehensive review of deep learning in EEG-based emotion recognition: Classifications, trends, and practical implications. *PeerJ Comput. Sci.* **2024**, *10*, e2065. [CrossRef]
33. Del Pup, F.; Zanolà, A.; Tshimanga, L.F.; Bertoldo, A.; Finos, L.; Atzori, M. The role of data partitioning on the performance of EEG-based deep learning models in supervised cross-subject analysis: A preliminary study. *Comput. Biol. Med.* **2025**, *196*, 110608. [CrossRef] [PubMed]
34. Liu, H.; Yang, S.; Zhang, Y.; Wang, M.; Gong, F.; Xie, C.; Liu, G.; Liu, Z.; Liu, Y.-J.; Lu, B.-L.; et al. Libeer: A comprehensive benchmark and algorithm library for eeg-based emotion recognition. *IEEE Trans. Affect. Comput.* **2025**, *16*, 3596–3613. [CrossRef]
35. Quan, J.; Li, Y.; Wang, L.; He, R.; Yang, S.; Guo, L. EEG-based cross-subject emotion recognition using multi-source domain transfer learning. *Biomed. Signal Process. Control* **2023**, *84*, 104741. [CrossRef]
36. He, Z.; Zhong, Y.; Pan, J. An adversarial discriminative temporal convolutional network for EEG-based cross-domain emotion recognition. *Comput. Biol. Med.* **2022**, *141*, 105048. [CrossRef] [PubMed]
37. Lara-Benítez, P.; Carranza-García, M.; Luna-Romera, J.M.; Riquelme, J.C. Temporal convolutional networks applied to energy-related time series forecasting. *Appl. Sci.* **2020**, *10*, 2322. [CrossRef]
38. Alizadegan, H.; Malki, B.R.; Radmehr, A.; Karimi, H.; Ilani, M.A. Comparative study of long short-term memory (LSTM), bidirectional LSTM, and traditional machine learning approaches for energy consumption prediction. *Energy Explor. Exploit.* **2025**, *43*, 281–301. [CrossRef]
39. Smith, L.N. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*; IEEE: New York, NY, USA, 2017; pp. 464–472.
40. Sartipi, S.; Cetin, M. Adversarial discriminative domain adaptation and transformers for EEG-based cross-subject emotion recognition. In *2023 11th International IEEE/EMBS Conference on Neural Engineering (NER)*; IEEE: New York, NY, USA, 2023; pp. 1–4.

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.

Article

PHASE: Progressive Hierarchical Adaptation for Sample-Efficient Rebalancing in Long-Tail Classification

Jiale Li ¹, Jicong Duan ^{1,2}, Changbin Shao ¹ and Hualong Yu ^{1,2,*}

¹ School of Computer, Jiangsu University of Science and Technology, Zhenjiang 212003, China; jiale_li@stu.just.edu.cn (J.L.); jicong_duan@stu.just.edu.cn (J.D.); shaocb@just.edu.cn (C.S.)

² School of Automation, Jiangsu University of Science and Technology, Zhenjiang 212100, China

* Correspondence: yuhualong@just.edu.cn; Tel.: +86-511-84409018

Abstract: When trained on long-tailed distributions, deep neural networks often suffer performance degradation and model bias due to the dominance of head classes. Existing reweighting and sampling strategies have significant limitations, such as reliance on fixed heuristics and inability to adapt to dynamic sample difficulty and class imbalance. Additionally, they fail to integrate sample-level granularity with class-level balance, further inadequately addressing the global imbalance issue. Motivated by these challenges, we introduce the Progressive Hierarchical Adaptation for Sample-Efficient rebalancing (PHASE) training framework, which employs a double-layer tuning paradigm to optimize performance under long-tailed distributions. Specifically, the double-layer tuning paradigm adopted by PHASE runs as follows: (1) an early-stage difficulty-aware mechanism targets those difficult-to-classify samples to guide representation learning; and (2) a later-stage multi-scale reweighting strategy integrates class distribution statistics with sample characteristics. This method ensures fine-grained adaptability and global balance, and thus outperforming those static or localized techniques. Extensive experiments on CIFAR10-LT, CIFAR100-LT, and ImageNet-LT datasets demonstrate that PHASE can significantly improve the accuracy of tail classes without degenerating head class performance, and acquire state-of-the-art classification results. PHASE provides a novel paradigm for long-tailed image recognition.

Keywords: long-tailed distribution; class imbalance; sample reweighting; difficulty-aware learning; double-layer tuning; representation learning

1. Introduction

In recent years, deep neural networks have made significant advancements in various computer vision tasks, such as image classification, object detection, and semantic segmentation. These advancements have been largely attributed to the availability of meticulously annotated [1–3] and well-balanced benchmarks like ImageNet [4], COCO [5], and Places [6]. These datasets play a crucial role as they provide a solid training foundation with evenly distributed classes and ample samples, facilitating the learning of comprehensive and transferable feature representations by the models. However, real-world data typically exhibit a long-tailed distribution, where a small number of classes (head classes) contain the majority of samples, leaving the remaining classes (tail classes) significantly under-represented. This class imbalance leads to a bias in training, where standard models tend to prioritize learning the dominant head classes, resulting in skewed decision boundaries and declined performance on the under-sampled tail classes. This bias poses a particular challenge in critical domains like medical diagnosis [7] and autonomous driving [8], where

misclassification or neglect of rare yet crucial categories can have severe consequences. To tackle this issue, researchers have explored various strategies, including advanced data augmentation methods, class-aware resampling techniques, and specific loss function designs. The objective of these approaches is to enhance recognition rates for minority classes while maintaining overall accuracy levels.

The most commonly utilized technique for addressing class imbalance in machine learning is resampling, which involves adjusting the class distributions in training data through either under-sampling the majority classes or over-sampling the minority classes. Despite its popularity, resampling has notable limitations: under-sampling can diminish the diversity and generalization capabilities of the majority classes, while over-sampling may result in overfitting the minority classes [9].

In contrast to resampling, reweighting presents a more direct and computationally efficient solution by assigning varying weights to the training loss of each class, thereby adjusting their relative importance during training. Traditional reweighting methods typically employ fixed weight allocation strategies based on the sample sizes of each class [10]. However, these static weights do not meet the dynamic training requirements of learning models, thereby constraining the adaptability and generalization of the model.

Motivated by aforementioned issue, we present a novel dynamic reweighting framework called Progressive Hierarchical Adaptation for Sample Effective rebalancing (PHASE) which is specifically designed to meet the challenges associated with long-tailed data distributions. PHASE adaptively tunes the weights of both head and tail classes throughout the entire training process for ensuring that each class is sufficiently represented. By dynamically balancing the contributions of both majority and minority classes, PHASE effectively lowers the inherent biases in long-tailed datasets, further enhancing overall performance and robustness of the learning model. Extensive experiments on three benchmark long-tailed image recognition datasets demonstrate that the proposed PHASE framework significantly improves recognition accuracy for tail classes, and meanwhile, improves the generalization of the overall model. It successfully overcomes the limitations of traditional resampling and static reweighting techniques, and provides a robust and extended solution for treating long-tailed data learning issues.

Three main contributions of this study can be summarized as follows:

- (1) By combining sample-level difficulty tuning with class-level weight allocation, PHASE systematically reconfigures optimization strategies tailored for long-tailed scenarios. Theoretically, PHASE achieves dynamic adaptation of local weights and equilibrium of global weights, creating a multi-stage, multi-scale adaptive optimization system that enhances learning model performance on long-tailed datasets.
- (2) In the initial phase, PHASE incorporates a dynamic tuning factor utilizing sample prediction probabilities to establish a Difficulty-Aware Weighting (DAW) framework. This framework enables the adjustment of the learning impact of individual samples, enhancing the model's sensitivity to the intricate sample distributions of tail classes. It actively redirects the trajectory of representation learning, further constructing a more robust feature space. In the second phase, a Multi-Scale Reweighting (MSR) approach is introduced. MSR integrates statistical patterns obtained from class distributions with the intrinsic data characteristics of each class. This approach effectively merges local and global class weight assignments, dynamically balancing the optimization paths of head and tail classes. It alleviates the adverse effects of weight competition-induced negative transfer between classes.
- (3) Extensive experiments are performed on three commonly utilized long-tailed datasets: CIFAR10-LT, CIFAR100-LT, and ImageNet-LT. The results indicate that PHASE outperforms existing state-of-the-art long-tailed reweighting approaches in terms of both

overall classification accuracy and tail class recognition accuracy. Moreover, when integrated with data augmentation methods, PHASE achieves better or comparable accuracy than/to those of the current state-of-the-art techniques, indicating its efficacy and superiority in addressing tasks involving imbalanced data distributions.

The rest of this paper is organized as follows. Section 2 briefly reviews some previous work closely associated with this study, including reweighting methods and decoupling representation techniques. In Section 3, we describe the proposed method in detail. Section 4 provides the experimental results and the corresponding analysis to verify the effectiveness and superiority of the proposed method. Finally, Section 5 concludes this study.

2. Related Work

2.1. Reweighting

Reweighting methods generally involve two different types: loss weighting and logit adjustment. Loss weighting balances the model's focus between head and tail classes by assigning different loss weights to samples belonging to different classes. Such methods include Class-Balance Loss [11], which uses the number of samples as weights, and Focal Loss [12], which emphasizes those difficult-to-learn samples by adjusting weights based on prediction probabilities. Logit adjustment techniques [13], on the other hand, allocate larger margins to tail classes, further alleviating imbalance. Additionally, methods such as Label-Distribution-Aware Margin (LDAM) loss [14], which enforces class-dependent margins, and Vector-Scaling (VS) loss [15], which combines additive and multiplicative logit adjustments, have also been proposed. These reweighting strategies effectively reduce bias towards head classes and improve classification performance of tail classes.

2.2. Decoupling Representation

Recent advancements in long-tailed visual recognition have emphasized using decoupling representation learning to enhance performance. The first method in such type, LDAM-DRW [14], adopts a two-phase training strategy, in which the first phase concentrates on learning robust feature representations, and the second phase employs delayed reweighting to refine decision boundaries. This approach significantly improves long-tailed prediction accuracy, although its underlying mechanisms could not be fully understood at that time. The foundational work on decoupled training [16] introduced this double-stage framework, and explained how it is different from traditional end-to-end paradigms. In the initial phase, various sampling strategies are empirically tried to search the most effective one for representation learning. In the second phase, the feature extractor is first fixed, and then different training strategies are evaluated and compared. Results indicate that the best choice is to use random sampling for optimizing representation learning and classifier re-adjustment as the training strategy, as this combination can significantly improve long-tailed recognition performance.

Based on these findings, MiSLAS [17] has been proposed. It adopts data augmentation by mix-up techniques in the first phase to strengthen representation learning, and employs a label-aware smoothing strategy in the second phase to improve classifier generalization. The improvements on both stages help to yield superior classification performance on long-tailed datasets. More recent studies have further refined the classifier training phase to alleviate specific challenges induced by long-tailed distributions. For example, SimCal [18] proposed a double-layer class balancing sampling strategy for dealing with long-tailed instance segmentation issue. It is helpful for calibrating the classification head and reinforcing the classifier training procedure.

These strategies have consistently shown superior classification performance on the aforementioned long-tailed datasets. To more clearly situate our proposed PHASE frame-

work within the existing landscape and highlight its unique contributions, we provide a comparative summary of main characteristics between PHASE and several representative state-of-the-art methods [12,14,19,20] in Table 1.

Table 1. A comparative summary of main characteristics between PHASE and several representative state-of-the-art methods.

Method	Training Paradigm	Focus on Stage 1	Focus on Stage 2	Sample-Level Difficulty Awareness?	Intra-Class Characteristic Consideration?
Focal Loss [12]	Single-stage	N/A	N/A	Yes (via modulating factor)	No
LDAM-DRW [14]	Two-stage	Minimize class-adjusted margin-based loss	Reweight by class frequency	Indirectly (via margin)	No
AREA [19]	Two-stage	Standard cross-entropy loss	Reweight by “effective area” of classes	No	Yes (via class feature distribution)
SURE [20] (conceptually)	Single-stage	N/A	N/A	Yes (via loss reweighting)	Implicitly (via robust optimization)
PHASE (Ours)	Two-stage	Difficulty-Aware Weighting (DAW)	Multi-Scale Reweighting (MSR)	Yes (explicitly via DAW)	Yes (explicitly via MSR)

As illustrated in Table 1, while existing methods often focus on a single aspect—such as sample difficulty (Focal Loss [12], SURE [20]), class-level margins (LDAM-DRW [14]), or intra-class distribution (AREA [19])—the PHASE framework is uniquely designed to systematically and explicitly integrate these complementary strategies in a progressive manner.

Unlike LDAM-DRW [14], which applies a static margin and delayed reweighting, PHASE introduces a dynamic, sample-level difficulty-aware weighting in the first stage to guide a more robust representation learning.

Unlike AREA [19], which uses a fixed feature distribution for reweighting in the second stage, PHASE proposes a multi-scale reweighting strategy that combines global class imbalance statistics with local, dynamic intra-class characteristics.

This progressive hierarchical adaptation—first refining features by sample difficulty, then refining the classifier by both class and intra-class balance—constitutes the novel contribution of PHASE, enabling a more fine-grained and effective handling of long-tailed distributions.

3. Methods

In this section, we introduce PHASE that is a novel approach designed to address long-tailed distributions issue in visual recognition tasks. Specifically, PHASE comprises two core components: DAW and MSR. Section 3.1 describes the primary challenges of long-tailed learning, followed by detailed descriptions of DAW in Section 3.2 and MSR in Section 3.3.

3.1. Preliminaries

The most critical challenge in long-tailed learning lies in effectively learning under inherently imbalanced class distributions. Consider a training set $D = \{(x_i, y_i)\}_{i=1}^N$ composed of N samples, where each sample x_i associates with a label $y_i \in \{1, 2, \dots, C\}$. If the dataset exhibits a highly imbalanced class distribution $(N_1, N_2, \dots, N_C)$, where N_j denotes the number of training instances in the j -th class, and typically the classes with fewer samples are less represented than those with more samples, it is considered a long-tailed dataset. This imbalance poses significant challenges to conventional training paradigms. The large discrepancy in the number of samples across different classes leads to

representation biases, making learning models focus more on the head classes, but neglect the tail classes. To address the core challenges in long-tailed learning outlined above, we propose the PHASE framework. It is built on a progressive hierarchical adaptation paradigm, whose core idea is to decouple and sequentially optimize two critical aspects: robust feature representation learning and balanced classifier calibration.

3.2. Difficulty-Aware Weighting Framework

We propose DAW framework that prioritizes those challenging instances during representation learning. Inspired by asymmetric focal-like paradigms [21], the DAW framework reduces the weights of easily classified samples and meanwhile, emphasizes those more difficult ones, thereby providing valuable information for later training. By adaptively tuning weights based on sample recognition difficulty level, it lays a solid foundation for further weight optimization at second stage, as illustrated in Figure 1.

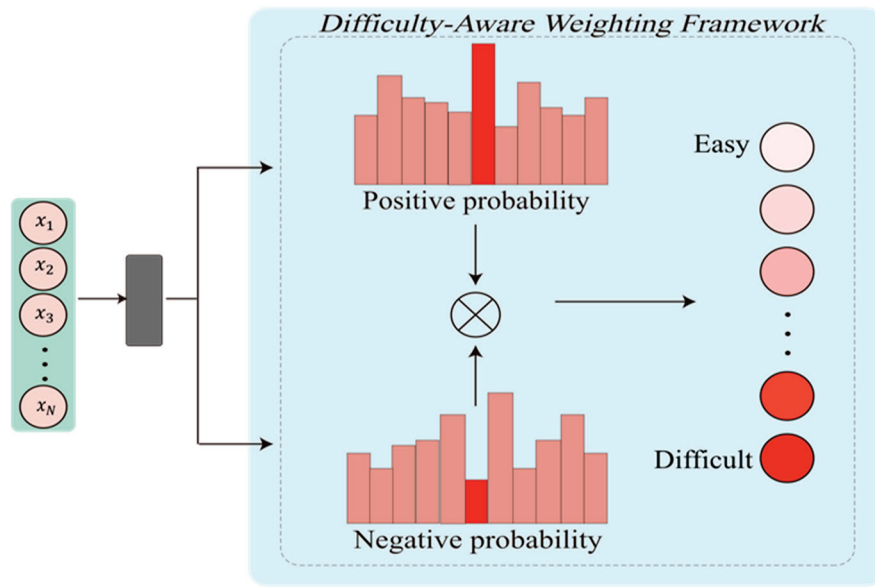


Figure 1. The difficulty-aware module dynamically guides representation learning to construct a robust feature space.

The model predicts a probability distribution $p(x_i) = [p_1(x_i), p_2(x_i), \dots, p_c(x_i)]^T$ satisfying the condition of $\sum_i p_j(x_i) = 1$, where $p_j(x_i)$ denotes the predicted probability that x_i belongs to the class j . Then it can be used to calculate the following asymmetric loss ('ASL'):

$$L_{ASL} = -\frac{1}{N} \sum_{i=1}^N \left[w_i^+ \log p_{y_i}(x_i) + \sum_{j \neq y_i} w_{i,j}^- \log(1 - p_j(x_i)) \right] \quad (1)$$

where w_i^+ and $(w_{i,j}^-)$ represent dynamic weights for positive and negative samples, respectively. These weights can be defined as:

$$w_i^+ = (1 - p_{y_i}(x_i))^{\gamma^+}, \quad w_{i,j}^- = (p_j(x_i))^{\gamma^-} \quad (2)$$

where γ^+ and γ^- are hyper-parameters which are used to control the degree of gradient decay. The positive sample weight w_i^+ decreases with the increase in $p_{y_i}(x_i)$, which can be used to lower the contribution of easily classified samples, while the negative sample weight $w_{i,j}^-$ reduces for small $p_j(x_i)$, which can be used to avoid excessive penalties on

well-separated negative samples. To unify the gradient tuning mechanism for both positive and negative samples, we introduce a weight correction function $A(x_i, j)$ as follows:

$$A(x_i, j) = (1 - p_j(x_i))^{\gamma^+ \cdot \mathcal{I}(j = y_i) + \gamma^- \cdot \mathcal{I}(j \neq y_i)} \quad (3)$$

where $\mathcal{I}(\cdot)$ is an indicator function that is used to distinguish positive samples ($j = y_i$) and negative samples ($j \neq y_i$). With $A(x_i, j)$, the ASL Loss can be rewritten as:

$$L_{\text{ASL}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C A(x_i, j) \cdot t_{i,j} \cdot \log p_j(x_i) \quad (4)$$

where $t_{i,j}$ is the one-hot label distribution of x_i . To further improve the model's robustness to tail-class samples and label noise, we also adopt the label smoothing technique to transform the one-hot label $t_{i,j}$ into a smoothed distribution $t'_{i,j}$ as follows:

$$t'_{i,j} = \begin{cases} 1 - \epsilon + \frac{\epsilon}{C}, & \text{if } j = y_i \\ \frac{\epsilon}{C}, & \text{if } j \neq y_i \end{cases} \quad (5)$$

where $\epsilon \in [0, 1]$ is the smoothing parameter that is used to balance the probabilities between the real class and other classes. By incorporating label smoothing, the final loss function can be expressed as:

$$L_{\text{final}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C A(x_i, j) \cdot t'_{i,j} \cdot \log p_j(x_i) \quad (6)$$

The optimization objective for the first stage is then defined as:

$$\Theta^{(1)} = \underset{\Theta}{\operatorname{argmin}} \mathbb{E}_{(x,y) \sim D_{\text{train}}^{(1)}} [L_{\text{final}}(x, y; \Theta)] \quad (7)$$

where $D_{\text{train}}^{(1)}$ represents the training dataset used at the first stage. After finishing the training of the first stage, the model parameters $\Theta^{(1)}$ have been optimized, and would be then fixed to provide stable feature representations for the second stage.

The complete procedure of the proposed ASL-based Difficulty-Aware Weighting (DAW) is summarized in Algorithm 1.

Algorithm 1: DAW

Input: Training set $D_{\text{train}}^{(1)} = \{(x_i, y_i)\}_{i=1}^N$, the number of classes C , the positive and negative focusing parameters γ^+ , γ^- , the label smoothing parameter ϵ

Output: Optimized feature extractor parameters $\Theta^{(1)}$

Procedure:

1. Initialize the model with γ^+ , γ^- and ϵ ;
 2. Define the log-softmax function for output normalization;
 3. For each training instance (x_i, y_i)
 - (a) Calculate predicted class probabilities $p(x_i)$ using softmax;
 - (b) Generate one-hot label vector $t_{i,j}$;
 - (c) Calculate asymmetric weights $A(x_i, j)$ as defined in Equation (3);
 - (d) Apply label smoothing to obtain smoothed labels $t'_{i,j}$ according to Equation (5);
 4. Aggregate the instance-wise losses to obtain the final loss L_{final} ;
 5. Optimize model parameters $\Theta^{(1)}$ by minimizing the expected loss on $D_{\text{train}}^{(1)}$ as defined in Equation (7).
-

3.3. Multi-Scale Reweighting

In long-tailed distributions, relying solely on static reweighting based on sample quantities always fails to adequately capture the complex intra-class feature distributions. To address this issue, we propose a Multi-Scale Reweighting (MSR) method and conduct it at the second stage of training. Specifically, MSR combines global weights tuning based on sample quantities with local dynamic weights tuning based on the effective area [19], which is illustrated in Figure 2.

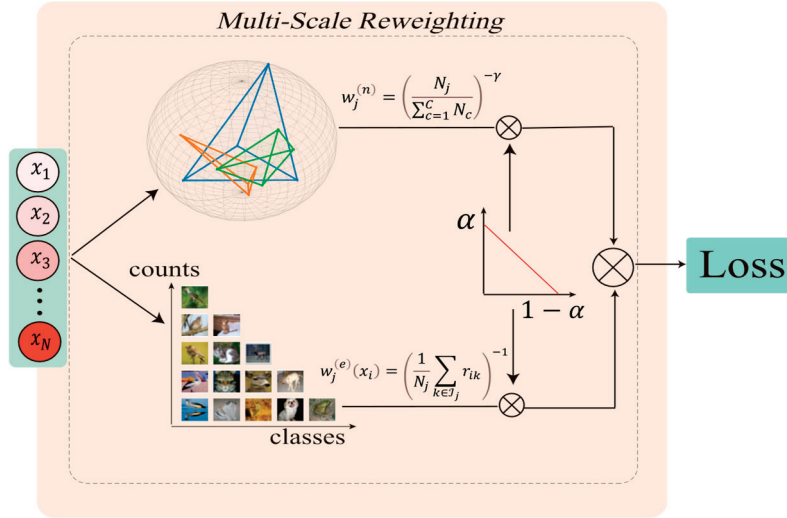


Figure 2. The weight allocation mechanism combines class distribution statistics with sample characteristics to balance the optimization between head and tail classes.

Both tunings are conducted at the class level to dynamically determine the integrated weight of each sample for enhancing the learning quality of tail classes. First, to address the imbalance issue existing in sample quantities among classes, we define a global tuning weight $w_j^{(n)}$ to balance the contributions of different classes. Let N_j denote the number of samples in class j , and C represent the number of classes. The global tuning weight can be defined as follows:

$$w_j^{(n)} = \left(\frac{N_j}{\sum_{c=1}^C N_c} \right)^{-\gamma} \quad (8)$$

where $\gamma > 0$ is a hyperparameter that is used to control the sensitivity of the weight distribution to sample quantities. This formulation is inspired by the concept of effective number of samples [11]. The parameter γ governs the trade-off between alleviating head-class bias and maintaining training stability. As $\gamma \rightarrow 0$, $w_j^{(n)} \rightarrow 1$, the loss reduces to a standard balanced loss, treating all classes equally but failing to address the imbalance. As $\gamma \rightarrow \infty$, the weight for the smallest class becomes excessively large, which may lead to overfitting on tail classes and instability. Thus, an appropriate γ is chosen to smoothly amplify the gradient contributions of tail classes during backpropagation without causing divergence. Here, classes with fewer samples (i.e., tail classes) are generally assigned larger weights to amplify their contributions to the overall optimization. Furthermore, the global weight should be normalized as:

$$\bar{w}_j^{(n)} = \frac{w_j^{(n)}}{\sum_{c=1}^C w_c^{(n)}} \cdot C \quad (9)$$

Although the global tuning alleviates inter-class imbalance, it cannot fully capture the diversity within each class. To model the discrepancy among intra-class distributions,

we further introduce a local dynamic tuning mechanism based on the similarity among samples within each class. Using the feature extractor that has been fixed at the first stage, we compute the feature representation of a sample x_i , denoted as $\varphi(x_i) \in R^d$. For a specific class j , the class feature center μ_j is calculated by:

$$\mu_j = \frac{1}{N_j} \sum_{x_i \in \mathcal{J}_j} \varphi(x_i) \quad (10)$$

where $\mathcal{J}_j = \{i \mid y_i = j\}$ is the set of indices for indicating which samples belong to the class j . Based on the feature deviation $\Delta\varphi(x_i) = \varphi(x_i) - \mu_j$, the similarity between a sample x_i and another sample x_k within the same class can be calculated as follows:

$$r_{ik} = \frac{\Delta\varphi(x_i)^\top \Delta\varphi(x_k)}{\|\Delta\varphi(x_i)\| \cdot \|\Delta\varphi(x_k)\|} \quad (11)$$

where $\|\cdot\|$ denotes the Euclidean norm. By using these similarity scores, the local dynamic tuning weight for the sample x_i is defined as:

$$w_j^{(e)}(x_i) = \left(\frac{1}{N_j} \sum_{k \in \mathcal{J}_j} r_{ik} \right)^{-1} \quad (12)$$

This equation highlights those samples which deviate more significantly from the class center or have weaker relationships with other samples, making the model better handle the diverse feature distribution within each class. To combine global tuning and local dynamic tuning, we define the integrated class-level weight $w_j(x_i)$ as:

$$w_j(x_i) = \alpha \cdot \bar{w}_j^{(n)} + (1 - \alpha) \cdot w_j^{(e)}(x_i) \quad (13)$$

where $\alpha \in [0, 1]$ is a tradeoff factor that controls the relative contributions of global and local weight tunings. The convex combination in Equation (13) is crucial for stability. The factor α balances trust between global class-level statistics and local sample-level characteristics. Setting $\alpha = 1$ relies solely on robust global statistics but may ignore intra-class diversity. Setting $\alpha = 0$ focuses entirely on intra-class similarity, which might be sensitive to feature noise. A value $0 < \alpha < 1$ harnesses the complementary strengths of both. Importantly, this formulation ensures the integrated weight $w_j(x_i)$ is bounded, i.e., $\min(w_j^{(n)}, w_j^{(e)}) \leq w_j(x_i) \leq \max(w_j^{(n)}, w_j^{(e)})$, preventing gradient explosion or vanishing that could arise from uncontrolled weight scaling, thereby enhancing training stability. At the second stage of training, the model uses the integrated weight $w_j(x_i)$ to optimize the classifier. The objective function is defined as:

$$\Theta^{(2)} = \arg \min_{\Theta} \frac{1}{N} \sum_{i=1}^N w_{y_i}(x_i) \cdot L_{CE}(x_i, y_i; \Theta) \quad (14)$$

where $L_{CE}(x_i, y_i; \Theta)$ represents the cross-entropy loss. The loss function in Equation (14) is a weighted empirical risk minimization. Since the feature extractor is fixed in stage two, the weights $w_{y_i}(x_i) > 0$ are positive and bounded scalars determined prior to classifier training. Given that the cross-entropy loss L_{CE} is smooth and differentiable with respect to the classifier parameters θ , the gradient of L_{MSR} remains Lipschitz continuous. Therefore, when optimized using standard stochastic gradient descent (SGD) with a suitable learning rate schedule (as in Section 4.1), the training process is expected to converge to a stationary point [22,23], consistent with the convergence guarantees of the underlying optimizer. The role of MSR is to reshape the loss landscape for more balanced learning without altering these fundamental convergence properties. By integrating the global weight tuning based

on sample quantity with the local weight tuning based on intra-class similarity, MSR can effectively enhance the learning quality of tail classes, and meanwhile preserve the diversity within class distributions, helping achieve a more balanced optimization for long-tailed distributions.

The procedure of the proposed MSR algorithm is summarized in Algorithm 2.

Algorithm 2: MSR

Input: Training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$; the sample counts of each class $\{N_j\}_{j=1}^C$; the feature extractor $\varphi(\cdot)$; the hyper-parameter $\gamma > 0$; the tradeoff factor $\alpha \in [0, 1]$

Output: The integrated class-level weights $\{w_j(x_i)\}_{i=1}^N$

Procedure:

1. Calculate the global tuning weight $w_j^{(n)}$ for each class $j \in \{1, 2, \dots, C\}$ by using Equation (8) and normalize it via Equation (9);
 2. Obtain the fixed feature representations $\varphi(x_i)$ for all training samples by using the pre-trained feature extractor;
 3. For each class j , calculate its class center μ_j by using Equation (10);
 4. For each instance x_i with class label $y_i = j$:
 - (a) Compute the feature deviation $\Delta\varphi(x_i) = \varphi(x_i) - \mu_j$
 - (b) For each sample x_k in the same class, calculate their cosine similarity r_{ik} by using Equation (11);
 - (c) Calculate local dynamic weight $w_j^e(x_i)$ by using Equation (12);
 5. Calculate the integrated weight $w_j(x_i)$ by using Equation (13);
 6. Use $w_j(x_i)$ to reweight cross-entropy loss during the second-stage classifier training via Equation (14);
 7. Output final sample weights $\{w_j(x_i)\}_{i=1}^N$.
-

4. Experiments

In this section, we conducted a comprehensive evaluation of the performance of the proposed PHASE framework on three widely used long-tailed image classification benchmark datasets: CIFAR10-LT, CIFAR100-LT, and ImageNet-LT. These datasets exhibit varying degrees of class imbalance and serve as standard benchmarks for assessing the robustness and generalization capabilities of long-tailed learning approaches. To gain a deeper understanding of the specific contributions of each component within the PHASE framework, we also performed extensive ablation studies. These analyses enable us to independently assess the impact of individual modules and investigate the interactions between different components, offering valuable insights into the internal mechanisms and design decisions of the PHASE framework. Furthermore, we conducted comparative evaluations against several state-of-the-art long-tailed learning methods. The comparative results show the efficacy of our proposed approach across datasets with diverse levels of class imbalance, thereby confirming the superior performance of PHASE in addressing real-world long-tailed recognition tasks.

4.1. Experiment Settings

To assess our proposed approach thoroughly, we initially generated various long-tailed versions from both CIFAR-10 and CIFAR-100 datasets using an exponential decay sampling method [24] applied to the original balanced datasets. An imbalance factor ($IF = \frac{N_{\max}}{N_{\min}}$) was defined, in where $N_{\max}$ and $N_{\min}$ represent the sample counts in the most and least represented classes, respectively. By adjusting the IF, we could precisely manage the degree of imbalance in the resulting dataset. Subsequently, CIFAR10-LT (Figure 3a) and

CIFAR100-LT (Figure 3b) each produced five training subsets with varying imbalance ratios in $\{200, 100, 50, 20, 10\}$. Furthermore, we utilized ImageNet-LT (Figure 3c), a subset of the ImageNet dataset comprising 200 classes and 100,000 images resized to 64×64 pixels. ImageNet-LT demonstrates an imbalance factor of 256, providing a more demanding benchmark for assessing the resilience and generalization capabilities of the proposed method. Detailed information regarding these long-tailed datasets is available in Table 2.

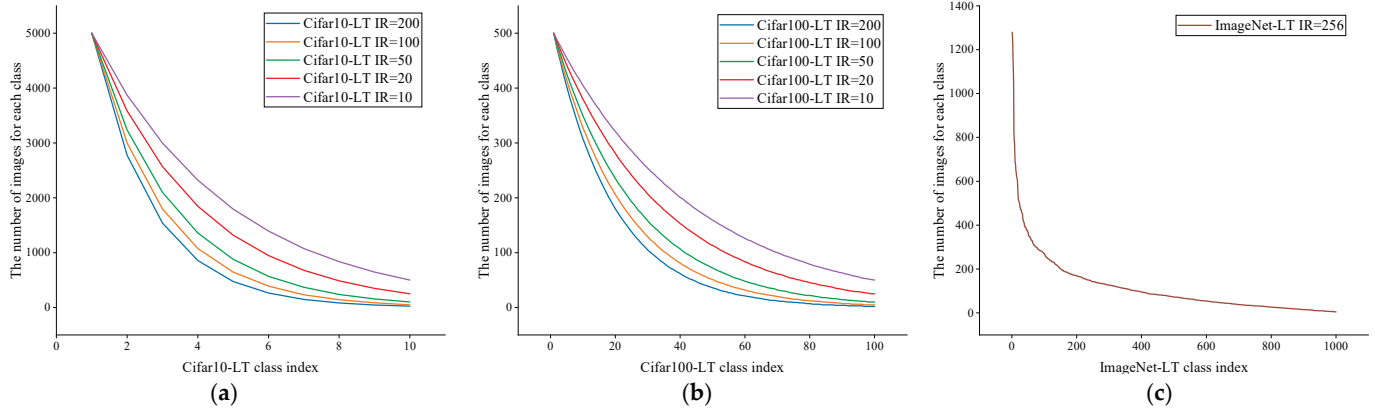


Figure 3. Class-wise image distributions of three long-tailed benchmark datasets: (a) Cifar10-LT, (b) Cifar100-LT, and (c) ImageNet-LT. Each curve represents the number of images per class under a specific imbalance ratio (IR). The figure illustrates the severity of class imbalance, where head classes contain significantly more samples than tail classes.

Table 2. The detailed information of the used long-tailed datasets.

Dataset	Imbalance Ratio	Number of Training Samples	Number of Testing Samples	Number of Classes	Size of Max Class	Size of Min Class
CIFAR10-LT	200, 100, 50, 20, 10	11,203~50,000	10,000	10	5000	25~500
CIFAR100-LT	200, 100, 50, 20, 10	9502~50,000	10,000	100	500	2~500
ImageNet-LT	256	115,846	50,000	1000	1280	5

In our experiments, all models were trained using the PyTorch (v2.0.1) framework [22] on a GeForce RTX 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA). We utilized the ResNet-32 architecture as the backbone for training on both CIFAR10-LT and CIFAR100-LT datasets, following the configuration specified in [19]. The models were trained for 200 epochs using standard SGD with a momentum of 0.9 and weight decay 5×10^{-4} . The initial learning rate was 0.1 with a linear warmup schedule [25], which was subsequently reduced to 1×10^{-3} at epoch 160 and further decreased to 1×10^{-5} at epoch 180 to facilitate convergence. For experiments on the ImageNet-LT dataset, we employed ResNet-50 as the backbone, with an initial learning rate of 0.1 and weight decay of 2×10^{-4} . To ensure fair comparisons with baseline methods, consistent data augmentation techniques such as random horizontal flipping and random cropping were applied to all models.

4.2. Long-Tailed Benchmark Results

To validate the effectiveness and superiority of our proposed method, we conducted comprehensive comparative experiments on multiple long-tailed image recognition datasets. Given the rapid advancements in this field, we selected various state-of-the-art methods for comparison, including single-stage reweighting methods and double-stage decoupled representation approaches. For instance, the Class-Balanced Loss method [11] adjusts the loss function by inversely weighting it based on the effective number of samples, thus ensuring a balanced training influence across different classes. In contrast, methods like LDAM-DRW [14] and AREA [19] employ a double-stage training approach, which

consists of initial training followed by subsequent rebalancing, aiming to improve classification performance on long-tailed datasets. These strategies have consistently shown superior classification performance on the aforementioned long-tailed datasets.

It is notable that methods utilizing double-stage decoupled representation strategies exhibit a closer association with our PHASE approach. We conducted extensive experiments on the CIFAR10-LT and CIFAR100-LT datasets to assess the performance of our method, PHASE, in comparison to several state-of-the-art baseline approaches [11,12,14,19,26,27], across five different imbalance factors (10, 20, 50, 100, 200). Table 3 displays the Top-1 accuracy of each approach on these datasets. The results of our experiments demonstrate that class-based reweighting techniques, such as DRW [14] and AREA [19], notably improve model accuracy regardless of the degree of class imbalance when combined with DAW. Particularly, integrating DAW with DRW and/or AREA yields substantial enhancements in accuracy, underscoring the efficacy of the proposed DAW component.

Table 3. Comparison results on CIFAR10/100-LT datasets. Top-1 accuracy (%) are reported, with the highest accuracy on each dataset have been highlighted in bold.

Dataset	CIFAR-10					CIFAR-100				
IF	200	100	50	20	10	200	100	50	20	10
Cross-Entropy	65.68	70.36	74.81	82.23	86.39	34.84	38.32	43.85	51.14	55.71
Focal [12]	65.29	70.38	76.71	82.76	86.66	35.62	38.41	44.32	51.95	55.78
LDAM [14]	-	73.35	-	-	86.96	-	39.6	-	-	56.91
L2RW [26]	66.25	72.23	76.45	81.35	82.12	33.00	38.90	43.17	50.75	52.12
CB [11]	68.89	74.57	79.27	84.36	87.49	36.23	39.60	45.32	52.59	57.99
LDAM-DRW [14]	73.52	77.03	81.03	-	88.16	38.91	42.04	47.62	-	57.99
IB [27]	73.96	78.26	81.70	85.80	88.25	37.31	42.14	46.22	52.63	57.13
AREA [19]	74.99	78.88	82.68	85.99	88.71	43.85	48.83	51.77	57.02	60.77
Our methods										
DAW										
+DRW	74.81	78.73	82.53	86.13	88.94	44.46	48.96	52.52	57.67	61.10
+AREA	75.70	79.32	83.08	86.19	88.86	44.52	48.97	52.60	57.73	61.11
+MRS	75.93	79.37	83.18	86.30	89.04	44.84	49.07	52.79	57.76	61.23

Also, we observe that integrating DAW with MSR in our proposed PHASE method yields additional performance enhancements. In comparison to the leading reweighting method AREA, PHASE demonstrates accuracy improvements of {0.94%, 0.49%, 0.96%, 0.31%, 0.33%} on the CIFAR10-LT dataset and {0.99%, 0.24%, 1.02%, 0.74%, 0.46%} on the CIFAR100-LT dataset.

The consistent improvements observed in both datasets with all imbalance levels clearly verify that PHASE is superior to existing reweighting strategies. Besides enhancing overall accuracy, the detailed class-wise performance depicted in Figure 4 demonstrates that PHASE significantly improves the recognition accuracy of tail classes, which are often underrepresented in long-tailed distributions. It verifies the effectiveness of PHASE in addressing class imbalance and facilitating balanced feature learning, indicating that it is particularly suitable for long-tailed visual recognition tasks.

Next, an extensive assessment of PHASE was conducted to compare its performance with current state-of-the-art baseline methods on the ImageNet-LT dataset. The results in Table 4 demonstrate that PHASE achieves a Top-1 accuracy of 49.96% on the ImageNet-LT dataset, outperforming all existing reweighting techniques. To comprehensively evaluate the impact of PHASE on long-tailed distributions, we implemented a categorization approach based on the number of training samples as suggested in [28]. Specifically, classes in the ImageNet-LT dataset were categorized into three groups: head classes (more than

100 samples per class), medium classes (20 to 100 samples per class), and tail classes (fewer than 20 samples per class). Subsequently, the average accuracy was calculated for each group to assess whether the PHASE model can specifically enhance the performance of tail classes.

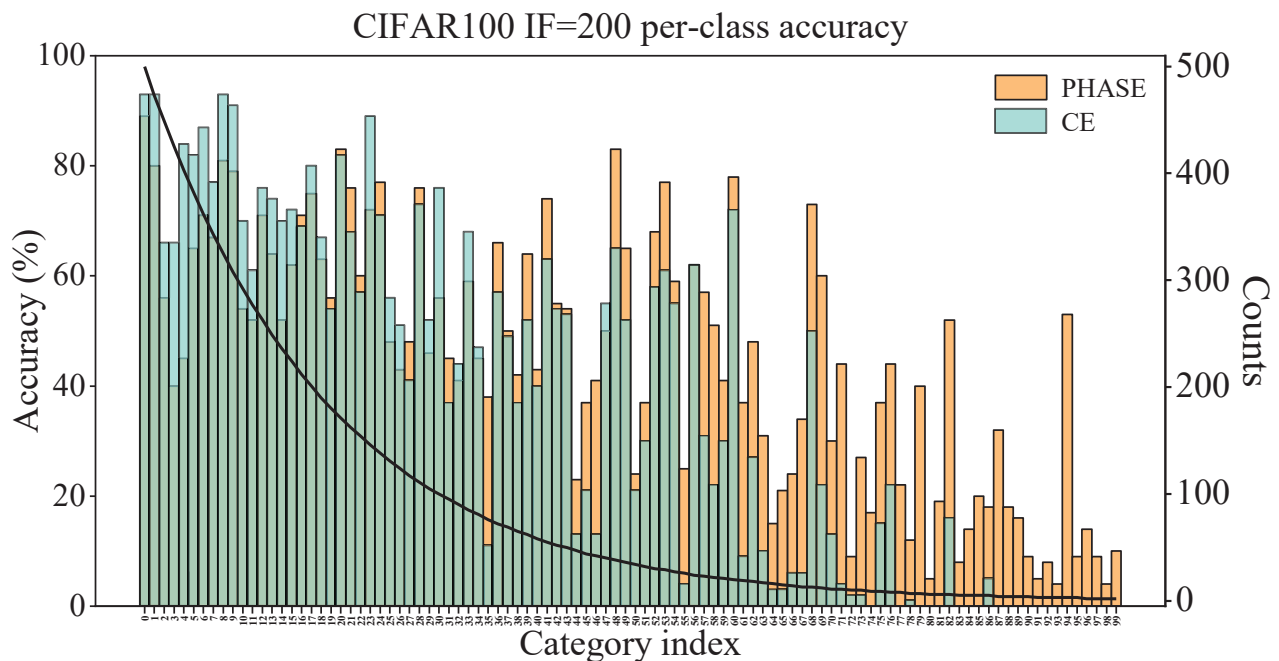


Figure 4. Classification accuracy and sample count of each class on CIFAR100-LT dataset (IF = 200).

Table 4. Comparison results of various methods on ImageNet-LT dataset, in where the highest accuracy has been highlighted in bold.

Method	Top-1 Accuracy
Cross-Entropy	38.88
Focal Loss [12]	30.50
Class-Balanced Loss [10]	40.85
LDAM [14]	41.86
LDAM-DRW [14]	45.74
Decoupling [16]	47.30
VS + TLA + ADRW [29]	48.21
PHASE	49.96

The results presented in Figure 5 demonstrate that PHASE consistently outperforms the double-stage method VS + TLA + ADRW [29] by 1.14%, 4.22%, and 6.78% in terms of Top-1 accuracy for head, medium, and tail classes, respectively. Additionally, PHASE shows significant superiority over the CE baseline by 16.10% and 27.74% in terms of Top-1 accuracy for medium and tail classes, with a minor decrease of 4.36% for head classes. These results suggest that PHASE effectively enhances the performance of tail and medium classes without compromising the performance of head classes, indicating its efficacy and superiority in managing long-tailed distribution data.

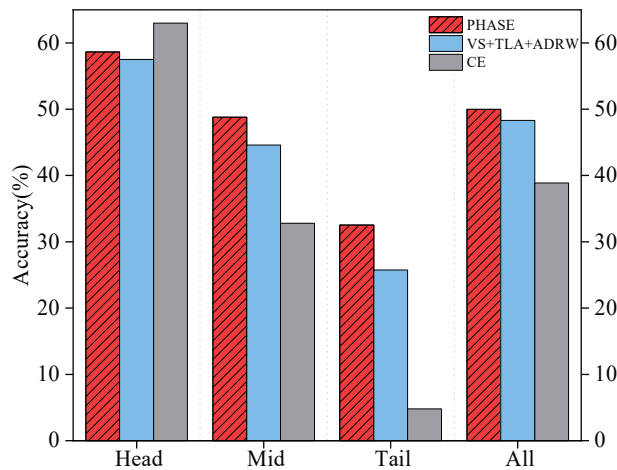


Figure 5. Comprehensive results on ImageNet-LT dataset.

4.3. Ablation Study

To rigorously assess the effectiveness of the proposed PHASE method, we conducted a series of thorough ablation experiments using the CIFAR100-LT dataset with an imbalance factor of 200. These experiments were carefully designed to systematically evaluate the individual contributions of each component within the PHASE framework. By analyzing each component both independently and in combination, the ablation study aims to deepen our understanding of the specific functional roles and interactions between DAW and MSR in addressing the class imbalance issue. This method enables us to not only determine the impact of each module on performance enhancement but also to explore how their integration results in synergistic benefits that surpass the capabilities of each component in isolation. The outcomes of this analysis further confirm the robustness, modularity, and efficacy of the PHASE framework, indicating its relevance to long-tailed visual recognition tasks characterized by highly skewed class distributions.

In Equation (13), we introduced a hyper-parameter, denoted as α , to balance the trade-off between two factors, $w_j^{(n)}$ and $w_j^{(e)}$. To assess the impact of α on model performance, we present the accuracy–weight variation curve in Figure 6. In this curve, α varies from 0 to 1 with an increment of 0.1. The results depicted in Figure 6 indicate that the performance initially improves with an increase in α . However, once it reaches an optimal value, the performance starts to decline. This observation aligns with our expectation that both weights are crucial for improving the performance of the model. Furthermore, we find that an optimal α setting can significantly improve the performance of the model. Notably, when α is set to 0.8, it enhances the performance by 0.15% and 0.32% compared to using $w_j^{(n)}$ and $w_j^{(e)}$ independently, respectively.

As previously outlined, the PHASE framework comprises two main modules: DAW and the MSR. To assess the efficacy of these two components individually and in combination, we conducted an ablation study using the CIFAR100-LT dataset with an imbalance factor of 200. The results, detailed in Table 5, indicate that both DAW and MSR, when employed separately, significantly enhance classification performance compared to the baseline approach. Furthermore, the integration of these two components within the PHASE framework leads to additional performance improvements, highlighting a synergistic relationship between the two strategies. These results not only confirm the effectiveness of each component in mitigating class imbalance challenges, but also verify the overall robustness of the unified PHASE framework in addressing long-tailed visual recognition tasks.

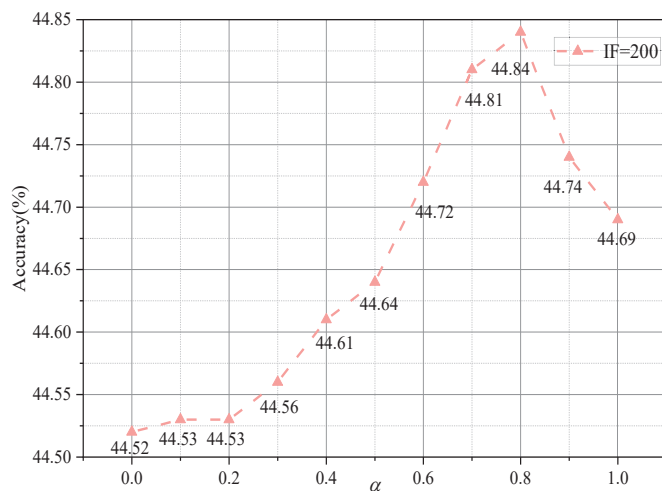


Figure 6. Impact of the parameter α at classification accuracy on CIFAR-100-LT (IF = 200) dataset.

Table 5. Ablation results of two main components of PHASE, where the best classification accuracy (%) on CIFAR100-LT (IF = 200) dataset has been highlighted in bold. Specifically, $\checkmark$ and $\times$ denote the corresponding component is involved or not, respectively.

DAW	MSR	Accuracy (%)
$\times$	$\times$	34.79
$\checkmark$	$\times$	37.93
$\times$	$\checkmark$	44.02
$\checkmark$	$\checkmark$	44.84

4.4. Comparison with SOTA Methods

In our initial experiments, we utilized a ResNet-32 architecture with 16 filters in the initial convolutional layer. However, recent state-of-the-art (SOTA) methods [20,30] suggest that employing deeper ResNet-32 architectures with 64 filters in the initial layer enhances classification performance. To ensure a fair comparison with these SOTA methods and fully maximize the potential of our proposed model, we also implemented a ResNet-32 architecture with the same number of channels. Furthermore, we integrated the Sharpness-Aware Minimization (SAM) technique [23] to enhance the optimization capability for tail classes. This technique assists in steering clear of saddle points and reaching flatter minima, thereby improving the overall performance [31].

In addition to the optimization techniques mentioned above, we also implemented various advanced image augmentation techniques [32], such as random cropping, horizontal flipping, and color jittering. These techniques aim to enhance input diversity, thereby enhancing the generalization ability of learning models in practical settings. We still conducted experiments on CIFAR100-LT dataset, with IF in range of {10, 50, 100}, respectively. The results presented in Table 6 demonstrate that the enhanced PHASE framework can achieve classification accuracy that is equal to or better than those state-of-the-art (SOTA) methods [20,30,33–35], thus confirming its efficacy and superiority. Specifically, PHASE performs best on CIFAR100-LT with IF = 100, and takes the second places on two other IF settings (only performs a little worse than SURE [32]). The results show that PHASE is robust enough as it performs excellent even if the class imbalance level is excessively high.

Table 6. Compared results with current SOTA methods on CIFAR100-LT dataset with IF in range of {10, 50, 100}, in where the highest accuracy on each IF setting has been highlighted in bold.

Method	IF = 100	IF = 50	IF = 10
GML [33]	54.00	58.10	67.00
DeiT-LT [34]	55.60	60.50	-
MDCS [35]	56.10	60.10	-
GLMC [30]	57.11	62.32	72.33
SURE [20]	57.34	63.13	73.24
PHASE	58.00	62.51	72.44

5. Conclusions

In this paper, we introduced PHASE, a novel progressive hierarchical adaptation framework for sample-efficient rebalancing in long-tailed classification. PHASE is designed around a two-stage tuning paradigm that systematically addresses both sample-level difficulty and class-level imbalance.

The effectiveness of PHASE is strongly demonstrated by our extensive experimental results. On CIFAR-100-LT with a high imbalance factor (IF) of 200, PHASE achieved a top-1 accuracy of 44.84%, significantly outperforming the state-of-the-art AREA method (43.85%). More notably, on the large-scale ImageNet-LT dataset, PHASE reached a top-1 accuracy of 49.96%, surpassing the previous best reweighting techniques and demonstrating a substantial improvement over the CE baseline (38.88%). A detailed analysis revealed that these gains were particularly pronounced for medium and tail classes, confirming PHASE's capability to alleviate model bias without compromising head class performance. The consistent superiority of PHASE across multiple benchmarks and imbalance ratios underscores the validity of its core design principle: decoupling difficulty-aware representation learning from multi-scale classifier rebalancing.

These findings highlight that adaptive, hierarchical tuning is a highly promising direction for long-tailed visual recognition. The PHASE framework provides a robust and effective baseline for future research in this area.

In future work, we plan to explore more efficient approximations for the MSR computation to enhance scalability. Furthermore, we will prioritize investigating the application of the PHASE paradigm to more complex and noisy real-world scenarios, as well as other challenging vision tasks suffering from data imbalance, such as object detection and semantic segmentation.

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

Funding: This study was partially supported by the National Natural Science Foundation of China under grant No.62176107.

Data Availability Statement: The datasets used in this study are available from the following sources: CIFAR10 and CIFAR100 are available at <https://www.cs.toronto.edu/~kriz/cifar.html> (accessed on 21 September 2025) and ImageNet is available at <https://image-net.org/> (accessed on 21 September 2025). The codes of the proposed PHASE algorithm can be freely downloaded from <https://github.com/Darren20000/PHASE> (accessed on 10 October 2025).

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

Abbreviations

The following abbreviations are used in this manuscript:

ASL	Asymmetric Loss
PHASE	Progressive Hierarchical Adaptation for Sample-Efficient rebalancing
DAW	Difficulty-Aware Weighting
MSR	Multi-Scale Reweighting
LDAM	Label-Distribution-Aware Margin
VS	Vector-Scaling
SGD	Stochastic Gradient Descent
SAM	Sharpness-Aware Minimization
SOTA	State-of-the-art

References

1. 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), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
2. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [CrossRef] [PubMed]
3. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
4. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
5. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
6. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1452–1464. [CrossRef] [PubMed]
7. Ayhan, M.S.; Kühlewein, L.; Aliyeva, G.; Inhoffen, W.; Ziemssen, F.; Berens, P. Expert-validated estimation of diagnostic uncertainty for deep neural networks in diabetic retinopathy detection. *Med. Image Anal.* **2020**, *64*, 101724. [CrossRef] [PubMed]
8. Choi, J.; Chun, D.; Kim, H.; Lee, H.-J. Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 502–511. [CrossRef]
9. Zhou, B.; Cui, Q.; Wei, X.-S.; Chen, Z.-M. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9719–9728.
10. Ren, J.; Yu, C.; Ma, X.; Zhao, H.; Yi, S. Balanced meta-softmax for long-tailed visual recognition. In Proceedings of the International Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020; pp. 4175–4186. [CrossRef]
11. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9268–9277. [CrossRef]
12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
13. Menon, A.K.; Jayasumana, S.; Rawat, A.S.; Jain, H.; Veit, A.; Kumar, S. Long-tail learning via logit adjustment. *arXiv* **2020**. [CrossRef]
14. Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 1567–1578.
15. Kini, G.R.; Paraskevas, O.; Oymak, S.; Thrampoulidis, C. Label-imbalanced and group-sensitive classification under overparameterization. In Proceedings of the International Conference on Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 18970–18983.
16. Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. *arXiv* **2019**. [CrossRef]
17. Zhong, Z.; Cui, J.; Liu, S.; Jia, J. Improving calibration for long-tailed recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 16489–16498. [CrossRef]

18. Wang, T.; Li, Y.; Kang, B.; Li, J.; Liew, J.; Tang, S.; Hoi, S.; Feng, J. The devil is in classification: A simple framework for long-tail instance segmentation. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; pp. 728–744. [CrossRef]
19. Chen, X.; Zhou, Y.; Wu, D.; Yang, C.; Li, B.; Hu, Q.; Wang, W. Area: Adaptive reweighting via effective area for long-tailed classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 19277–19287. [CrossRef]
20. Li, Y.; Chen, Y.; Yu, X.; Chen, D.; Shen, X. SURE: Survey recipes for building reliable and robust deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 17500–17510. [CrossRef]
21. Ridnik, T.; Ben-Baruch, E.; Zamir, N.; Noy, A.; Friedman, I.; Protter, M.; Zelnik-Manor, L. Asymmetric loss for multi-label classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 82–91. [CrossRef]
22. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8024–8035. [CrossRef]
23. Foret, P.; Kleiner, A.; Mobahi, H.; Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv* **2020**. [CrossRef]
24. Yue, C.; Long, M.; Wang, J.; Han, Z.; Wen, Q. Deep quantization network for efficient image retrieval. In Proceedings of the 13th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 3457–3463. [CrossRef]
25. Goyal, P. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv* **2017**. [CrossRef]
26. Ren, M.; Zeng, W.; Yang, B.; Urtasun, R. Learning to reweight examples for robust deep learning. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4334–4343. [CrossRef]
27. Park, S.; Lim, J.; Jeon, Y.; Choi, J.Y. Influence-balanced loss for imbalanced visual classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 735–744. [CrossRef]
28. Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; Yu, S.X. Large-scale long-tailed recognition in an open world. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2537–2546. [CrossRef]
29. Wang, Z.; Xu, Q.; Yang, Z.; He, Y.; Cao, X.; Huang, Q. A unified generalization analysis of re-weighting and logit-adjustment for imbalanced learning. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024; pp. 48417–48430. [CrossRef]
30. Du, F.; Yang, P.; Jia, Q.; Nan, F.; Chen, X.; Yang, Y. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 15814–15823. [CrossRef]
31. Rangwani, H.; Aithal, S.K.; Mishra, M. Escaping saddle points for effective generalization on class-imbalanced data. In Proceedings of the International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 22791–22805. [CrossRef]
32. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation policies from data. *arXiv* **2018**. [CrossRef]
33. Suh, M.K.; Seo, S.W. Long-tailed recognition by mutual information maximization between latent features and ground-truth labels. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 32770–32782. [CrossRef]
34. Rangwani, H.; Mondal, P.; Mishra, M.; Asokan, A.R.; Babu, R.V. DeiT-LT: Distillation strikes back for vision transformer training on long-tailed datasets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 23396–23406. [CrossRef]
35. Zhao, Q.; Jiang, C.; Hu, W.; Zhang, F.; Liu, J. MDCS: More diverse experts with consistency self-distillation for long-tailed recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 11597–11608. [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.

Article

Flow Field Evaluation Method of High Water-Cut Reservoirs Based on K-Means Clustering Algorithm

Chen Liu ^{1,2,3}, Qihong Feng ¹, Wensheng Zhou ^{2,3}, Chi Zhang ^{2,3} and Xianmin Zhang ^{1,*}

¹ School of Petroleum Engineering, China University of Petroleum (East China), Qingdao 266555, China; liu099339@163.com (C.L.); fengqihong.upc@gmail.com (Q.F.)

² State Key Laboratory of Offshore Oil and Gas Exploitation, Beijing 100028, China; zhouwsh@cnooc.com.cn (W.Z.); zhangchi20@cnooc.com.cn (C.Z.)

³ CNOOC Research Institute Ltd., Beijing 100028, China

* Correspondence: spemin@126.com

Abstract: In this paper, the concept of symmetry is utilized to evaluate the distribution characteristics of flow fields—that is, flow fields with balanced displacement generally exhibit good spatial symmetry. In the late stage of water-flooding reservoir development, identifying flow field distribution and implementing targeted adjustments are crucial for improving development efficiency and enhancing oil recovery. This study establishes a quantitative evaluation index system integrating both static geological and dynamic production factors to comprehensively characterize flow field distribution in ultra-high water-cut reservoirs. The system incorporates residual oil potential abundance, water-flooding ratio, and water influx intensity as key indicators. A flow field classification method based on the K-Means clustering algorithm was proposed, with the Davies–Bouldin index applied to evaluate clustering validity. The approach was validated using the Egg model, where the flow field was effectively classified into four types: inefficient retention field, effective displacement field, dominant displacement field, and extreme displacement field. Adjustment measures were then applied based on classification results. The findings demonstrate that the proposed method weakens dominant displacement areas while expanding effective and inefficient displacement zones, leading to a 1.1 percentage point increase in recovery factor. This research provides a practical and quantitative tool for flow field diagnosis and adjustment, offering valuable technical guidance for managing ultra-high water-cut reservoirs.

Keywords: ultra-high water-cut period; flow field; residual oil advantage potential abundance; K-Means clustering algorithm; Davies–Bouldin index

1. Introduction

Most mature oilfields in eastern China have entered high or ultra-high water-cut development stages. Long-term water flooding has significantly altered the properties of reservoir fluids, the pore structure of the formation, and the distribution of remaining oil [1]. The original reservoir heterogeneity has further intensified, leading to pronounced differences in oil–water flow capacities within the reservoirs. The underground flow field distribution exhibits significant heterogeneity, with dominant flow channels easily developing in local high-permeability zones and strong flow regions, which exacerbates the uneven utilization of crude oil in the reservoir. The problem of inefficient and ineffective water injection circulation is evident, significantly impairing the effectiveness of reservoir development [2,3]. Therefore, it is of great guiding significance for improving the recovery

rate to effectively identify the distribution of flow fields in ultra-high water-cut reservoirs and take targeted flow field adjustment measures [4].

To reflect the changing law of underground fluids, Chen (1998) first introduced the concept of flow field into petroleum engineering [5]. He described the state of fluid flow and proposed that the reservoir flow field not only serves as a storage space for reservoir fluids but also as the medium through which fluids move. Domestic and foreign scholars have carried out a large number of related studies on the identification and description of reservoir flow fields. Liu et al. (2010) proposed the concept of flow field intensity index by using the inter-well distribution factor and swept volume, and carried out the identification and evaluation of reservoir flow field based on the statistical analysis of the concept [6]. Chen (2011), Ma (2019), and others selected three parameters, namely water flow rate, fluid velocity, and water saturation, and quantitatively describe the reservoir flow field by coupling the analytic hierarchy process and fuzzy comprehensive evaluation method to calculate flow field intensity [7,8]. Bai et al. (2017) used pore volume, permeability, water saturation, and water-oil mobility ratio as the four key factors for the formation of the dominant flow field as evaluation factors, and established the discrimination standard of the dominant flow field through fuzzy comprehensive evaluation and streamline numerical simulation technology [4]. Zhang (2014) proposed to use the surface flux as the only parameter to characterize the flow field intensity and established the flow field intensity classification standard [9]. Lu et al. (2020) further selected the four parameters of permeability, formation pressure gradient, oil saturation, and surface flux as the main characterization parameters of the reservoir flow field, and established a quantitative characterization method of the reservoir flow field by combining the analytic hierarchy process and fuzzy mathematics theory [10]. The above flow field evaluation methods mostly use fuzzy comprehensive evaluation methods to evaluate the flow field intensity. This method has a lot of subjective factors and cannot objectively reflect the impact of each evaluation index on the reservoir flow field [10].

During development, reservoirs with obvious seepage differences should belong to different flow fields, and the same flow field has similar characteristics on water flooding and residual oil distribution [11]. Streamline numerical simulation is based on streamline theory, the most important features of which are high computational efficiency and flow field identification. Streamlines can more intuitively reflect the instantaneous flow field distribution and reservoir utilization conditions [11–13]. In recent years, scholars have conducted extensive research on flow field identification and characterization in high water-cut reservoirs. By extracting spatial trajectories and attribute data from streamlined simulation results, multidimensional quantitative indicator systems for reservoir flow fields have been established. Combined with clustering, machine learning, and dimensionality reduction techniques, these approaches have achieved hierarchical identification and quantitative evaluation of complex reservoir flow fields. Jia and Deng [14] constructed a streamlined feature database by extracting spatial positions and flow property parameters from streamlined simulation results. Using artificial intelligence methods based on stream-lined clustering, they effectively identified and classified flow field types in high water-cut reservoirs, overcoming the subjectivity and boundary ambiguity of traditional methods. Building upon this, Guo et al. [15] proposed a flow field optimization method based on artificial intelligence algorithms. By analyzing the hierarchical features of streamlined fields, they dynamically optimized and adjusted displacement effect partitions within reservoirs. Zhao et al. [16] developed a vector flow field characterization and reconstruction method based on streamline simulation. By considering the complex seepage characteristics of high water-cut reservoirs and analyzing streamlined trajectories and attributes, they achieved dynamic description and visualized displacement processes, effectively revealing dominant

flow channels and residual oil distribution patterns during late-stage water flooding. Li et al. [17], using unsupervised machine learning techniques, integrated multiple parameters such as streamline density, flow velocity, and pressure fields to objectively classify and hierarchically evaluate complex reservoir flow field types. Liu et al. [18] combined principal component analysis (PCA) with clustering algorithms to perform dimensionality reduction and feature extraction on high-dimensional, multi-variable flow field data, improving the computational efficiency for large-scale, high-dimensional reservoir flow field processing. Through clustering analysis, they completed flow field partitioning and identification, significantly enhancing the adaptability and computational efficiency of complex reservoir flow field characterization. The flow field evaluation method based on clustering algorithms comprehensively considers the dynamic percolation characteristics and the space physical characteristics of the reservoir, dividing the flow field with obvious differences in oil and water seepage well [11]. However, the above methods cannot fully reflect the underground flow field distribution of the actual reservoir, especially for a region with potential flow space but no fluid flow at a certain moment, as they only use the extracted characteristic parameters to streamline clusters. In addition, these methods cannot finely distinguish different types of flow fields, because they treat each streamline as a single sample point. Regions with obviously different flow capacities in the same streamline will be classified as one type. How to comprehensively and accurately distinguish the flow field of high water-cut reservoirs is still a challenging problem.

To address the above issues, this study fully considers the complexity of reservoir flow fields during the high water-cut stage, which are jointly controlled by static geological characteristics and dynamic seepage patterns. Using actual reservoir geological model grids as characterization units, a comprehensive set of multidimensional feature parameters is systematically extracted for each spatial grid cell, covering aspects such as the material basis of the flow field, displacement efficiency, and water-flooding status. A grid-scale flow field characterization index system is then established to comprehensively reflect the current development status and flow field characteristics of the reservoir. On this basis, an unsupervised clustering algorithm is introduced to develop a flow field characterization and evaluation method suitable for reservoirs in the high water-cut stage. Unlike traditional clustering approaches based on streamlined clusters, this method treats all geological model grid cells as independent samples and establishes a data-driven flow field partitioning approach based on multidimensional feature data. This enables effective identification of potential flow pathways and regions with differences in seepage capacity, overcomes misjudgments caused by streamlined cross-regional flow in conventional methods, and significantly improves the accuracy and rationality of reservoir flow field partitioning.

2. Construction of Flow Field Characterization Indices

The flow field of a reservoir is a complex system in which multiple factors influence and interact with each other. During the water-flooding development, the flow field will undergo uneven changes. It is difficult to comprehensively characterize and evaluate the flow field of an ultra-high water-cut reservoir using a single attribute factor [7]. The flow field characteristics of ultra-high water-cut reservoir are not only related to the static heterogeneity of the reservoir, but also to the dynamic heterogeneity of reservoir development [10]. Therefore, a comprehensive and quantitative characterization of the flow field is performed by taking into full consideration the static and dynamic flow field factors of an ultra-high water-cut reservoir, combined with the characteristics of severe reservoir heterogeneity, and constructing an evaluation index system from three dimensions: flow field material basis, displacement degree, and water-flooding status.

2.1. Residual Oil Advantage Potential Abundance

Affected by multiple geological and development factors, the underground seepage field of the ultra-high water-cut reservoir has become more uneven, and the reservoir heterogeneity and the dispersion of remaining oil have become more prominent. The uneven distribution of the flow field is the main reason for the dispersion of remaining oil and low injection-production efficiency. Saturation used to be the only indicator for flow field characterization, which can only reflect the flow field distribution or water-flooding state in the reservoir, but cannot quantify its potential.

Taking full account of the oil-water distribution and flow characteristics of the reservoir in the ultra-high water-cut period, the movable oil advantage potential abundance is introduced as one of the flow field potential evaluation indicators. The calculation formula is as follows:

$$VOI = \frac{\mu_w K_{ro}}{\mu_o K_{rw}} \cdot [S_o - S_{or}] \cdot \phi \cdot h \cdot NTG \quad (1)$$

where, VOI is the movable oil advantage potential abundance, mPa·s; μ_w is the formation water viscosity, mPa·s; μ_o is the formation oil viscosity, mPa·s; K_{ro} is the oil relative permeability; K_{rw} is the water relative permeability; S_o is the oil saturation; S_{or} is the residual oil saturation; ϕ is the porosity; h is the reservoir thickness, m; NTG is the net-to-gross ratio.

After long-term water injection and flushing, the reservoir physical properties, pore structure, fluid properties, and seepage characteristics of the ultra-high water-cut reservoir have undergone significant changes. The static and dynamic heterogeneity of the reservoir has further increased, resulting in a sharp increase in the difference in capacity in oil-water diversion [19]. A different water-flooding degree means a huge difference in oil-water diversion capacity. The semi-logarithmic relationship curve between the oil-water relative permeability ratio and water saturation no longer shows a completely linear relationship, but shows a downward bend [20,21]. In order to characterize the downward bend phenomenon of the oil-water relative permeability ratio curve in the ultra-high water-cut period, based on previous research [22], a new relative permeability curve characterization relationship is proposed:

$$\ln(K_{ro}/K_{rw}) = a + b \ln(S_w) + 2 \ln(c - S_w) \quad (2)$$

where, S_w is water saturation; a , b , and c are fitting parameters, and $c = 1 - S_{or}$ when the residual oil saturation S_{or} is known.

Four-phase permeability curve data of a block in the literature [21,23] were selected and fitted using the new characterization relationship (2). As shown in Figure 1, the four-phase permeability curves have an obvious downward bending. The new phase permeability characterization relationship (2) achieves high-precision fitting of the four-phase permeability curve data throughout the process, and the correlation coefficients R^2 reach 0.9989, 0.9997, 0.9983, and 0.9982, respectively, which verifies the accuracy of the new phase permeability characterization formula.

Substituting the new phase permeability curve characterization relationship (2) into Equation (1), we can get the following:

$$VOI = e^a \frac{\mu_w}{\mu_o} \cdot (c - S_w)^2 \cdot [S_o - S_{or}] \cdot S_w^b \cdot \phi \cdot h \cdot NTG \quad (3)$$

The movable oil advantage potential abundance can not only characterize the size of the movable remaining oil reserves underground, but also reflect the different oil-water diversion capacities in ultra-high water-cut reservoirs. The higher the movable oil advantage potential abundance index value, the greater the potential for reservoir flow field adjustment.

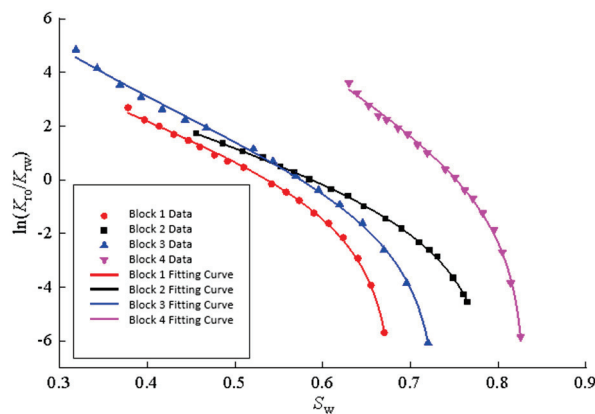


Figure 1. Curves of fitted new formulas in different blocks.

2.2. Water-Flooding Ratio

The water-flooding ratio is an important cumulative indicator reflecting the evolution of the flow field. It is defined as the volume ratio of accumulating injected water to unit pore volume. It is a concentrated reflection of the water drive intensity and the flushing volume of the injected water in the flow field [7]. After entering the high water-cut state, as the water-flooding ratio increases, the reservoir properties such as porosity, permeability, and mud content change, resulting in changes in the flow field [24]. Therefore, the water-flooding ratio is selected as an important indicator for flow field evaluation. The expression is

$$N_w = \left(\frac{W_i}{V_\phi} \right) = \left(\frac{W_i}{V_i \phi} \right) \quad (4)$$

where N_w is the water-flooding ratio; W_i is accumulated water injection, m^3 ; V_ϕ is the pore volume flowing through the cell, m^3 ; V_i is the volume flowing through the cell, m^3 .

Assuming that the flow of oil-water phases in the process of water-flooding oil complies with Darcy's law, and the effects of gravity and capillary force are not considered. The oil and water phase flow equation can be obtained:

$$q_w = q_o \frac{\mu_o K_{rw}}{\mu_w K_{ro}} \quad (5)$$

where, q_o is the oil flow rate, m^3/d ; q_w is the water flow rate, m^3/d .

Substituting Equation (1) into the above equation, we can get

$$q_w = q_o \frac{\mu_o}{\mu_w} e^{-a} S_w^{-b} (c - S_w)^{-2} \quad (6)$$

Integrating Equation (6) with respect to production time, we can get the cumulative water production of the stage:

$$W_p = \int_0^t q_w dt = e^{-a} \int_0^t q_o \frac{\mu_o}{\mu_w} S_w^{-b} (c - S_w)^{-2} dt \quad (7)$$

where, W_p is the cumulative water production of the stage, m^3 .

The cumulative oil production of the stage can be expressed as

$$N_p = N \left(1 - \frac{S_o}{1 - S_{wi}} \right) = \frac{N}{1 - S_{wi}} (S_w - S_{wi}) = V_\phi (S_w - S_{wi}) \quad (8)$$

where, N_p is the cumulative oil production of the stage, m^3 ; S_{wi} is the initial water saturation; N is the reserves of crude oil, m^3 .

Taking the derivative of both ends of Formula (8) with respect to time t , the relationship between oil phase flow rate and cumulative oil production of the stage is:

$$q_o = \frac{dN_p}{dt} = V_\phi \frac{dS_w}{dt} \quad (9)$$

Substituting Formula (9) into Formula (6), we can get

$$W_p = e^{-a} V_\phi \frac{\mu_o}{\mu_w} \int_{S_{wi}}^{S_w} x^{-b} (c-x)^{-2} dx \quad (10)$$

Further derivation:

$$W_p = e^{-a} V_\phi \frac{\mu_o}{\mu_w} \int_{S_{wi}}^{S_w} \left[\frac{1}{c^2 x^b} + \frac{2c-1}{c^2 (c-x)^2} \right] dx \quad (11)$$

Further integration and sorting can get the relationship between cumulative water production of the stage:

$$W_p = e^{-a} V_\phi \frac{\mu_o}{c^2 \mu_w} \left[\frac{S_w^{1-b} - S_{wi}^{1-b}}{1-b} + \frac{(2c-1)(S_w - S_{wi})}{(c-S_w)(c-S_{wi})} \right] \quad (12)$$

According to the material balance principle, when the injection-production balance is maintained, the cumulative water injection volume of a stage is equal to the sum of the cumulative oil production and the cumulative water production of the stage. Therefore, the cumulative water injection volume of a stage can be expressed as

$$W_i = W_p + B_o N_p \quad (13)$$

where, W_i is the cumulative water injection volume of the stage, m^3 ; B_o is the oil formation volume factor.

Substituting Equations (8) and (12) into Equation (13), we can get

$$W_i = e^{-a} V_\phi \frac{\mu_o}{c^2 \mu_w} \left[\frac{S_w^{1-b} - S_{wi}^{1-b}}{1-b} + \frac{(2c-1)(S_w - S_{wi})}{(c-S_w)(c-S_{wi})} \right] + B_o V_\phi (S_w - S_{wi}) \quad (14)$$

Substituting Equation (14) into Equation (4), we can get the following expression for calculating the water flow ratio:

$$N_w = e^{-a} \frac{\mu_o}{c^2 \mu_w} \left[\frac{S_w^{1-b} - S_{wi}^{1-b}}{1-b} + \frac{(2c-1)(S_w - S_{wi})}{(c-S_w)(c-S_{wi})} \right] + B_o (S_w - S_{wi}) \quad (15)$$

According to Equation (15), the water flow ratio is closely related to water saturation, and water saturation reflects the water-flooding effect. When the initial water saturation is constant, the water-flooding intensity is high in the area with high flow field intensity. The larger the water flow ratio, the smaller the residual oil saturation.

2.3. Water Influx

Water influx can be used to assess the relative water-oil flow capacity and sweep status in the reservoir. By analyzing the changes in water influx, the dynamic evolution characteristics of the reservoir flow field distribution can be captured. Without considering

the effects of capillary force and gravity, the calculation formula for water influx (i.e., water content) is as follows:

$$f_w = \frac{1}{1 + \frac{\mu_w K_{ro}}{\mu_o K_{rw}}} \quad (16)$$

where, f_w is the water influx.

Substituting Equation (2) into Equation (16), we can get

$$f_w = \frac{1}{1 + \frac{\mu_w}{\mu_o} e^a S_w^b (c - S_w)^2} \quad (17)$$

The increasing water influx represents the improvement of the flow capacity of injected water. The water phase occupies an increasingly higher proportion in the flow process and gradually becomes the dominant mobile phase in the underground flow field, further indicating the degree of water washing of the reservoir and the formation of a dominant flow field.

3. Establishment of Flow Field Evaluation Method

Taking the actual reservoir geological model grid as the characterization unit, the characteristic parameters of each grid are extracted according to the reservoir numerical simulation results. Including areas with potential flow space but no fluid flow, three evaluation indicators are calculated to comprehensively characterize the reservoir flow field distribution, such as the remaining oil potential abundance, water flow ratio, and water influx. The flow field characteristics of any spatial grid unit can be portrayed by these three indicators. The data need to be normalized to eliminate the dimensional and numerical range difference, which can influence the evaluation results. The remaining oil potential abundance is a positive indicator, while the water flow ratio and water influx are negative indicators. The normalization formula is as follows:

$$\overline{VOI} = \frac{VOI - VOI_{\min}}{VOI_{\max} - VOI_{\min}} \quad (18)$$

$$\overline{N_w} = \frac{N_{w\max} - N_w}{N_{w\max} - N_{w\min}} \quad (19)$$

$$\overline{f_w} = \frac{f_{w\max} - f_w}{f_{w\max} - f_{w\min}} \quad (20)$$

where, $\overline{VOI}$ is the normalized residual oil advantage potential abundance; $\overline{N_w}$ is the normalized water-flooding ratio; $\overline{f_w}$ is the normalized water influx; $VOI_{\max}$ and $VOI_{\min}$ are the maximum and minimum values of residual oil advantage potential abundance, respectively; $N_{w\max}$ and $N_{w\min}$ are the maximum and minimum values of water-flooding ratio, respectively; $f_{w\max}$ and $f_{w\min}$ are the maximum and minimum values of water influx, respectively.

For the actual reservoir model with n grid units, there is a sample set D formed by n data samples of the flow field. Every sample is composed of an attribute set $\{\overline{VOI}, \overline{N_w}, \overline{f_w}\}$ of normalized three evaluation indicators with the dimension of three.

Cluster analysis is a typical unsupervised learning technology that is used to divide data samples of unknown categories into several clusters according to the data's internal relationships. It has been successfully applied to data clustering problems in different fields [24]. Based on the flow field distribution represented by grid units, the clustering algorithm can identify different flow field areas with different characteristics, and classify them into different types. The flow field evaluation method for ultra-high water-cut reservoirs is established, providing effective guidance for the formulation of subsequent

flow field adjustment measures. As the most popular clustering algorithm, the K-Means clustering algorithm has an efficient clustering effect for large data sets. Compared with other clustering algorithms, it has been widely used due to its simple principle, good clustering effect, and high clustering efficiency [25,26].

Generally, the K-Means clustering algorithm selects Euclidean distance to measure the similarity between data samples, and uses the sum of squared error (SSE) as the objective function [27]. By minimizing the objective function, data samples are divided into k clusters according to their distance from the cluster center [26]. The core idea [25] is: (1) specify the initial number of clusters k ; (2) and randomly select k data samples from the data sample set as the initial cluster centers and calculate the Euclidean distance between other data samples and each cluster center; (3) find the cluster center closest to the target data sample, and assign the data sample to the cluster corresponding to the cluster center; (4) calculate the average value of the data samples in each cluster as the new cluster center and perform the next iteration; and (5) the clustering ends when the clusters no longer change or the objective function converges.

The Euclidean distance calculation formula between the data sample and the cluster center is

$$d(x, c_i) = \sqrt{\sum_{j=1}^3 (x_j - c_{ij})^2} \quad (21)$$

where, x is the data sample in the data sample set D , $x = \{\overline{VOI}, \overline{N_{S_w}}, \overline{f_w}\}$; c_i is the i -th cluster center, $1 \leq i \leq k$; x_j and c_{ij} are the j -th attribute values of the data sample x and the cluster center c_i , respectively.

SEE of an entire data set can be calculated by formulation:

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} |d(x, c_i)|^2 \quad (22)$$

where, C_i is the i -th data sample cluster; k is the number of clusters.

The flowchart of the flow field evaluation method for high water-cut reservoirs based on the K-means clustering algorithm is shown in Figure 2.

The number k of clusters is used to be determined in advance. However, it is difficult to determine the k value due to excessive data volume and lack of experience in practical applications. If the k value is too small, the data samples within the same cluster will be very different, making it difficult to capture the effective feature information hidden in the cluster data samples; if the k value is too large, the data samples between different clusters will not have much differences, increasing the difficulty of data analysis. Therefore, many scholars have proposed clustering effectiveness evaluation functions as a criterion for determining the optimal number of clusters [14,28]. This paper selects the Davies–Bouldin (DB) index [29] as the clustering effectiveness evaluation function. The k value corresponding to the minimum DB index is automatically searched as the optimal number of clusters within the possible range. The DB index is an indicator that evaluates clustering quality by the average distance ratio of clusters and the density within the cluster. It comprehensively considers the similarities inside each cluster and the differences among clusters. The smaller the value is, the better the clustering result reaches. Compared with the Silhouette coefficient and Calinski–Harabasz index [28], the Davies–Bouldin (DB) index exhibits lower sensitivity to outliers and isolated points under high-dimensional and multi-cluster conditions, effectively reducing the interference of extreme samples on clustering evaluation results and ensuring the stability and reliability of clustering outcomes. In particular, when dealing with large-scale, high-dimensional, strongly heterogeneous, and

complex-flow reservoir grid flow field feature datasets, the DB index can objectively and accurately measure the compactness and separation of clustering partitions. It provides a robust criterion for assessing the objectivity, rationality, and scientific validity of flow field partitioning results in highly heterogeneous reservoirs during the high water-cut stage.

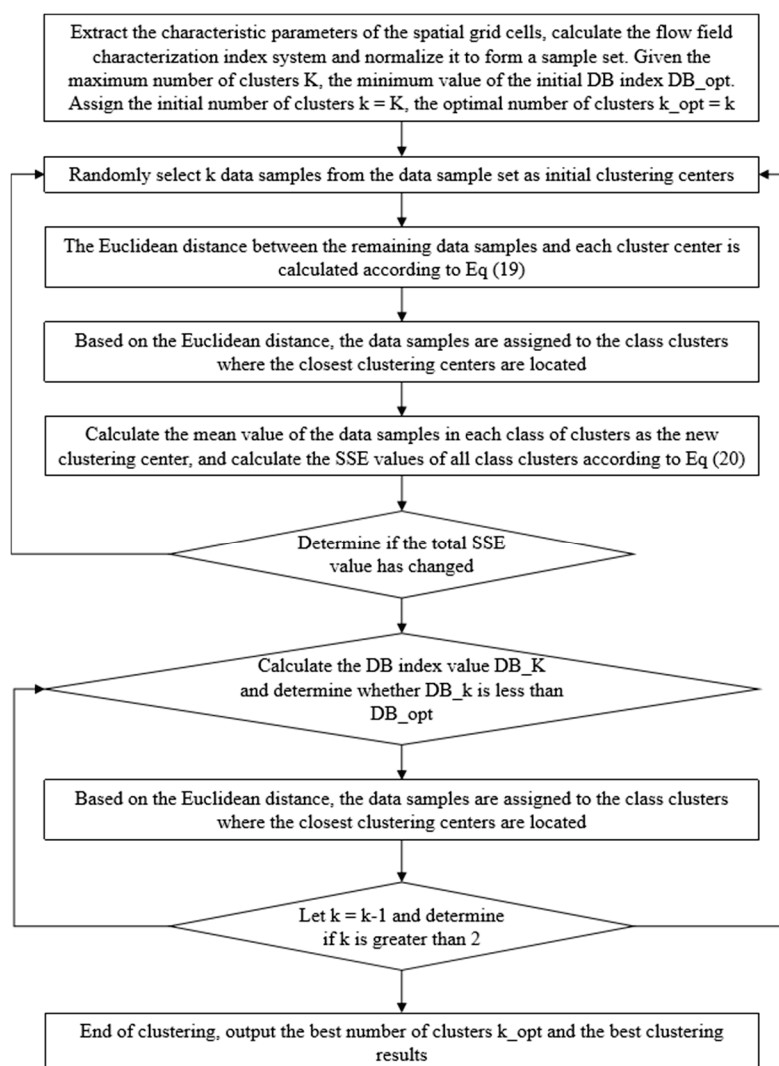


Figure 2. Flowchart of Flow Field Evaluation Method for High Water-Cut Reservoirs Based on K-Means Clustering Algorithm.

4. Case Study

The flow field evaluation method was used for testing and analysis on the Egg model [30,31] as the actual case. The Egg model is a small heterogeneous synthetic reservoir model. The model is an oil-water two-phase model without associated gas or dissolved gas. The grid dimension of the Egg model is $60 \times 60 \times 7$, with a total of 25,200 grids. The edge of the model is a dead grid, and there is a total of 18,553 valid grids. The permeability and well-location distribution of the Egg model are shown in Figure 3. The high-permeability area is distributed in a river channel shape. The relative permeability ratio curve of the Egg model is shown in Figure 4. The curve has an obvious downward bend. The curve fitting correlation coefficient R^2 reaches 0.9983, using the new relative permeability characterization relationship (2), which achieves high-precision fitting of the entire process. The fitting coefficients are $a = -3.61$, $b = -8.6366$, and $c = 0.8177$. The Egg model has a total of four production wells and eight water injection wells. After 10 years of simulated water injection

development, the recovery rate of the model reached 58.1%, and the comprehensive water content reached 97.5%, entering the ultra-high water-cut state.

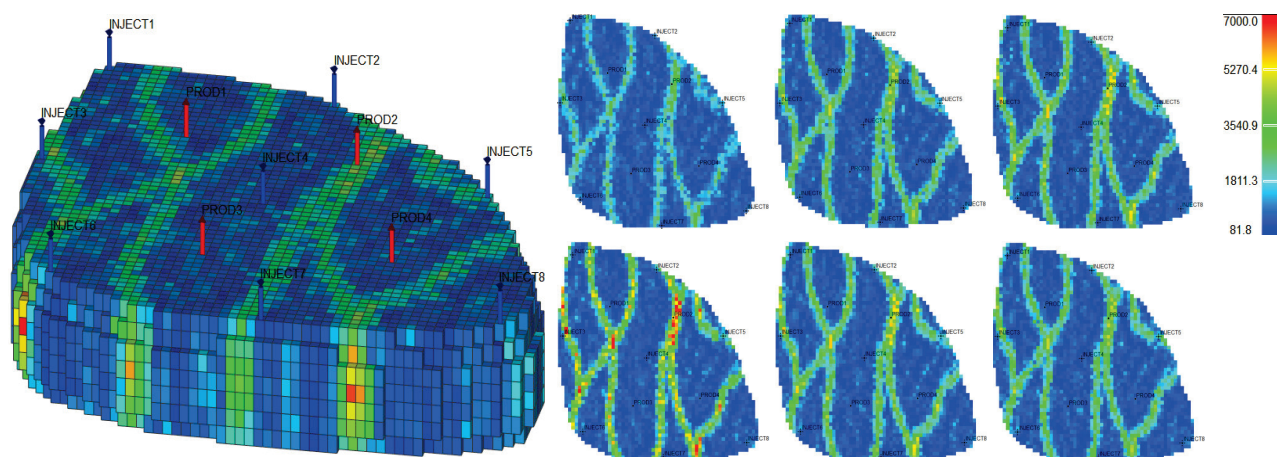


Figure 3. Permeability Distribution Map of the Egg Model.

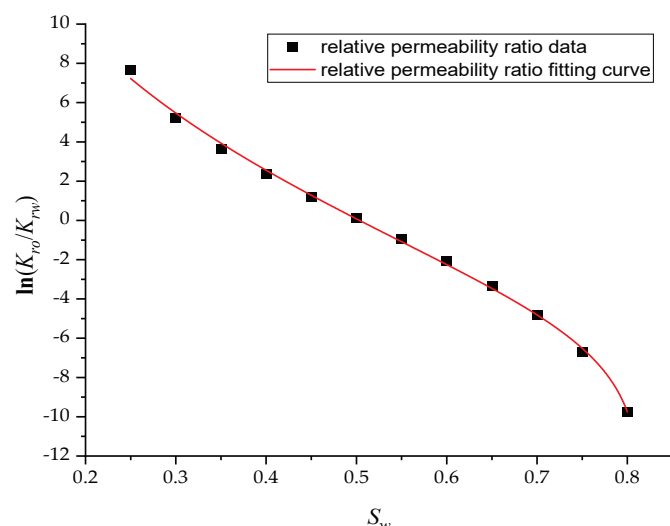


Figure 4. Curve fitting of phase permeability ratio for the Egg model.

4.1. Result

The 18,553 effective grids of the Egg model were used as characterization units, and the characteristic parameters of each grid were extracted. After 10 years of production and injection. Three flow field characterization indices, the residual oil advantage potential abundance, water-flooding ratio, and water influx, were calculated according to the numerical simulation results. To ensure that the clustering results possess good representativeness and interpretability, and to avoid excessive clustering leading to overly fine partitioning of flow field categories, which would weaken the practical value of flow field analysis, the number of clusters in this study is limited to no more than five. This constraint keeps the number of flow field partitions within a reasonable range, thereby facilitating subsequent flow field feature extraction and dominant zone identification. The relationship between the DB index values and the number of clusters is presented in Figure 5. Table 1 shows the flow field characterization and classification evaluation result of the ultra-high water-cut Egg model, using the novel flow field evaluation method. After entering the ultra-high water-cut stage, from the distribution of oil saturation and the abundance of remaining oil advantage potential shown in Table 1, the remaining oil retained underground still has great development potential. There are still some unaffected areas inside the oil layer, especially

the top layer with richer remaining oil and higher potential abundance, which is also the material basis for future flow field adjustment. The distribution of the water-flooding ratio shows that the water-flooding ratio is extremely large near the water injection well, as the driving intensity is large, while it is small when far away from the water injection well with a small water-flooding ratio. The displacement within and between layers is extremely unbalanced, and the water-flooding development can be affected by various geological and development factors. A single factor can only qualitatively analyze the displacement status and roughly judge the local potential area. It is hard to accurately identify and finely distinguish flow field areas with different characteristics through a single factor.

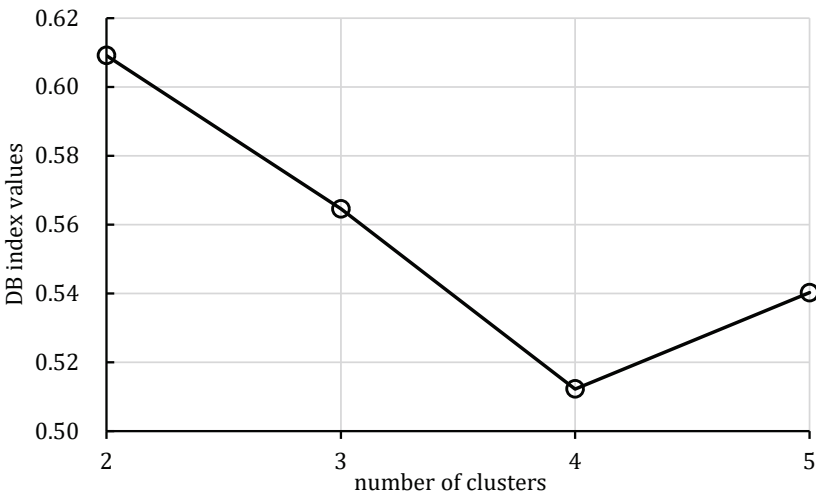
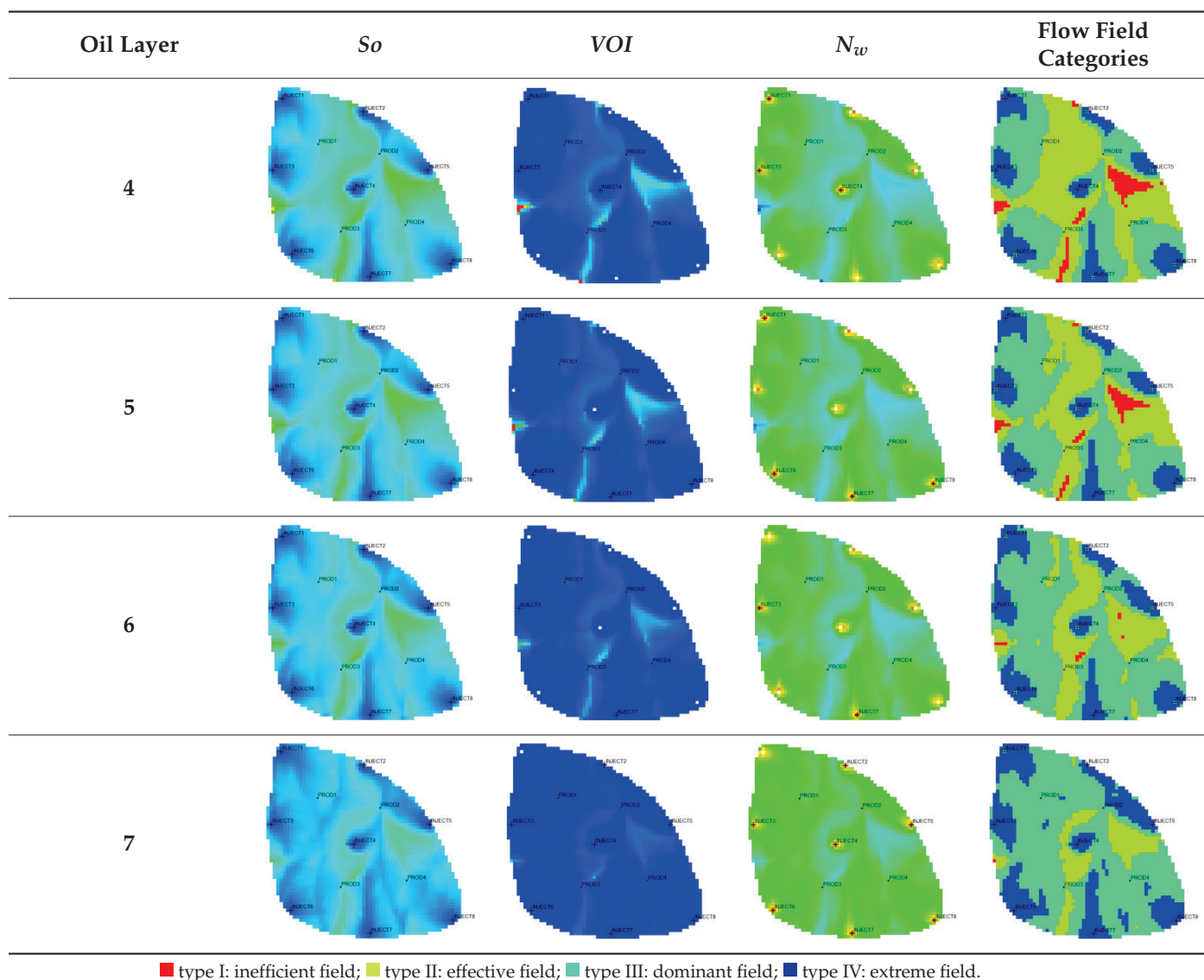


Figure 5. Relationship between the DB index values and the number of clusters.

Table 1. Reservoir flow field characterization and classification evaluation results for the Egg model.

Oil Layer	So	VOI	Nw	Flow Field Categories
1				
2				
3				

Table 1. Cont.



The new flow field evaluation method automatically identifies the flow field of the ultra-high water-cut Egg model into four types. Table 2 shows the four types of flow field clustering by cell attribute with its proportions. The flow field classified results of the Egg model simulated water injection development for 10 years show that the evaluation indices of different flow field types, like the regional remaining oil advantage potential abundance, water flow multiples, and water flow, have obvious differences. The new method captures the attribute characteristics of different flow field types accurately, and distinguishes the flow fields effectively, which is conducive to identifying flow field areas with development potential. According to the attribute characteristics with development potential, the flow field is divided into four types: inefficient field (type I), effective field (type II), dominant field (type III), and extreme field (type IV). Based on this, further analysis of different categories of flow fields was conducted using symmetry theory. Under ideal conditions of balanced displacement, the water-flooding flow field between injection and production wells exhibits a radially symmetrical distribution. However, under certain extreme heterogeneity conditions, effective displacement zones or inefficient displacement zones may disappear. Taking the flow field classification results of Sub-layer 1 in Table 1 as an example, the different types of flow field zones between injection well INJECT4 and production well PROD2 show a concentric ring-like symmetrical distribution, which,

from the core to the periphery, are arranged sequentially as extreme displacement zone, dominant displacement zone, effective displacement zone, and inefficient displacement zone, indicating an overall balanced displacement process. Between injection well INJECT1 and production well PROD1, the inefficient displacement zone disappears, while the other types of flow field zones display an approximately axisymmetric distribution, with a relatively balanced displacement. In contrast, under certain extreme heterogeneity conditions, such as between injection well INJECT7 and production well PROD3, the distribution of different types of flow field zones significantly deviates from the principal axis of the injection-production well pair. This is manifested as an abnormal deviation in the distribution structure of displacement capability zones or an attenuation of symmetry. Such zoning characteristics are closely related to reservoir heterogeneity features and the injection-production configuration, and the analysis results can provide theoretical support and technical guidance for water-flooding development adjustment and remaining oil potential tapping.

Table 2. Clustering attributes of different flow field categories.

Classes	Properties	Mean Residual Oil Advantage Potential Abundance	Mean Water-Flooding Ratio	Mean Water Influx	Field Proportion
	inefficient field	229.56	0.30	22.2%	10.0%
	effective field	18.66	0.47	46.3%	31.0%
	dominant field	6.66	0.60	62.9%	42.7%
	extreme field	2.09	0.82	77.1%	16.3%

From the identification and classification results of different types of flow fields in Tables 1 and 2, the flow field distribution of the reservoir is extremely complex after entering the ultra-high water-cut state, with the dominant field and the extreme field accounting for 59%, the effective field accounting for 31%, and the inefficient field accounting for 10%. The extreme field is mainly developed in the area near the injection wells and the local high-permeability zone between wells. For instance, the extreme fields existing between INJECT2, INJECT5, and production Wells PROD2, are formed due to long-term water flooding in high-permeability zones. The residual oil saturation in this type of flow field is extremely low, the mean water influx reaches 77.1%, the mean residual oil advantage potential abundance is only 2.09, and the development adjustment potential is low. Compared with the dominant field, the effective field has a lower regional average permeability, leading to a lower mean water-flooding ratio and mean water influx. However, the effective field has higher residual oil saturation and mean residual oil advantage potential abundance, contributing to higher development adjustment potential. The inefficient field is more complex in distribution, mainly near the diversion line area between injection and production wells, the local low permeability areas, or the top small layers. Injected water can be harder to spread but the residual oil is enriched, with mean residual oil advantage potential abundance reaching 229.56. It is the main area for potential exploitation of late flow field adjustment, as the abundance is much higher than the other three types. The high-potential areas are mainly distributed near the diversion line between the injection and production wells, and the flow field model is small. Although the dominant field and the effective field in the flow field model have a large area, there is little potential for well-infilling. As a result, the dominant field and extreme field are supposed to be restrained, while the inefficient field and effective field are supposed to be enhanced. Priority is given to adjusting the flow field through strategies like increasing injection, liquid lifting, injection and production optimization, profile control, and separate injection.

4.2. Discussion

According to the flow field distribution, the injection and production parameter optimization countermeasures are used to carry out the flow field adjustment study for the Egg model. The comparison of the working system of each well before and after the adjustment is shown in Figure 6. The figure shows that the water injection wells adopted a unified working strategy before the adjustment. After the injection and production parameter optimization countermeasures were adopted, the water injection wells in the area with a large local flow field intensity were liquid controlled, and the water injection of wells in the area with a large local flow field potential was increased. The same goes for the oil production wells.

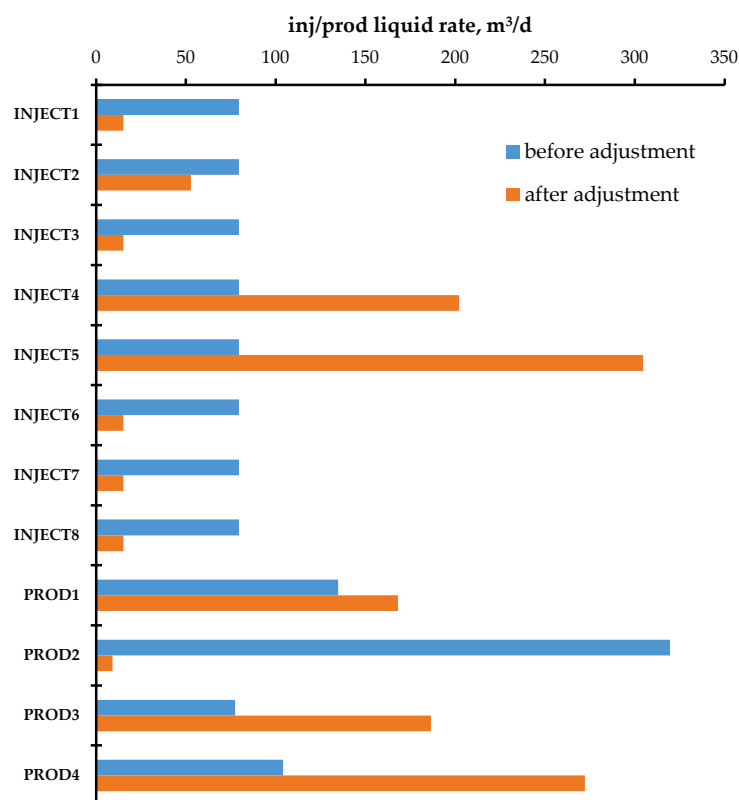
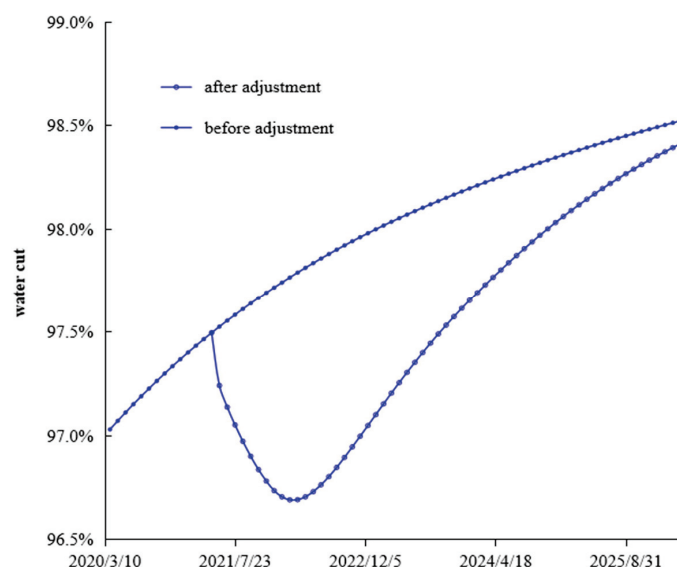


Figure 6. Comparison of fluid injection/production rates before and after flow field adjustment.

From the comparison table of clustering attributes of four types before and after flow field adjustment given in Table 3, after adjustment, the dominant field and extreme field areas were suppressed, and the effective field and inefficient field areas were strengthened. The original dominant flow field area was relatively weakened, and the proportion of the dominant field and extreme field areas decreased from 68% to 48.24%, and the proportion of the effective field and inefficient field areas increased from 25.76% and 6.23% to 43.18% and 8.57%, respectively. By comparing the flow field clustering attribute parameters before and after the adjustment, the average remaining oil dominant potential abundance of different flow field area types decreased significantly, which was only about 50% of that before the adjustment, indicating that the flow field adjustment effectively released the potential of different flow field types. Figure 7 shows a comparison of the comprehensive water content of the model after simulating production for 5 years before and after the flow field adjustment. After the flow field of the reservoir was adjusted during the ultra-high water-cut period, the development effect was significantly improved, the comprehensive water content decreased significantly, and the recovery factor increased by 1.1 percentage points.

Table 3. Comparison of clustering attributes for 4 flow field types before and after adjustment.

Classes	Mean Residual Oil Advantage Potential Abundance		Mean Water-Flooding Ratio		Mean Water Influx		Field Proportion	
	Before	After	Before	After	Before	After	Before	After
inefficient field	182.07	98.13	0.30	0.372	21.7%	32.6%	6.23%	8.57%
effective field	15.51	9.10	0.49	0.557	49.6%	58.2%	25.76%	43.18%
dominant field	5.49	3.54	0.63	0.700	65.7%	71.3%	45.08%	38.99%
extreme field	1.86	0.97	0.85	0.974	78.1%	82.9%	22.92%	9.25%

**Figure 7.** Comparison of comprehensive water cut before and after flow field adjustment.

The validation based on the Egg model demonstrates that the proposed method exhibits good adaptability and stability under different heterogeneity conditions and complex-flow field states. It provides a novel technical approach for the fine characterization of flow fields in highly heterogeneous reservoirs during the high water-cut stage. The research outcomes can offer reliable technical support for well pattern adjustment and optimization, chemical flooding deployment, and the optimization of remaining oil recovery strategies during reservoir development, possessing significant theoretical guidance and engineering application value. In particular, the method shows promising applicability and broad prospects in addressing challenges such as complex-flow field characteristics, pronounced differences in seepage capacity, and difficulties in identifying potential flow zones in highly heterogeneous reservoirs during high water-cut stages.

5. Conclusions

In this study, a novel flow field partitioning and characterization method for high water-cut reservoirs was proposed, addressing the complexity of flow field evolution jointly controlled by static geological properties and dynamic seepage behavior. By systematically extracting multidimensional characteristic parameters that reflect the material foundation of the flow field, displacement efficiency, and water-flooding status, a grid-scale flow field characterization index system was constructed to comprehensively depict the current development status and flow field characteristics of the reservoir.

Distinct from conventional methods based on streamline clustering or fuzzy logic—which rely heavily on subjective thresholds and are prone to misclassification due to streamlines crossing multiple regions—the proposed method treats all geological grid cells

as independent samples. It employs a K-Means clustering algorithm for objective flow field classification, with the Davies–Bouldin index used for cluster validity evaluation. This approach effectively overcomes limitations such as streamlining overlapping-induced misjudgment and clustering subjectivity, achieving fine-scale identification of different flow field types and accurate capture of flow field characteristics, thereby significantly improving the accuracy and rationality of flow field partitioning.

Validation using the benchmark Egg model demonstrated the method’s strong adaptability and stability under different heterogeneity conditions and complex-flow field scenarios. The approach successfully identified potential flow paths and regions with varying seepage capacities, classifying them into four flow field types: low-efficiency displacement, effective displacement, dominant displacement, and extreme displacement fields. Subsequent flow field adjustment based on these classifications led to a weakened dominant displacement region, enhanced effective and low-efficiency displacement zones, and an increase in overall oil recovery by 1.1 percentage points.

Compared with existing clustering-based flow field partitioning techniques, the proposed method leverages the advantages of multidimensional feature-driven analysis, avoiding the arbitrary threshold setting and subjectivity issues associated with fuzzy logic approaches, while also resolving the partitioning ambiguity and misclassification problems caused by streamline crossing in traditional streamline-based methods. This method is particularly suitable for high-dimensional, large-scale, and strongly heterogeneous reservoir grid flow field data analysis. It provides a robust, objective, and efficient technical tool for flow field identification and high water-cut reservoir development decision-making, offering significant theoretical value and practical application potential for well pattern adjustment, chemical flooding deployment, and remaining oil exploitation optimization in complex heterogeneous reservoirs.

While validated on synthetic models such as the Egg model, further testing on diverse real-field data sets is necessary to fully confirm its generalizability and practical utility. Future work will focus on addressing these limitations by exploring more advanced clustering algorithms, integrating multi-source reservoir data, and conducting extensive field validations.

Author Contributions: Conceptualization, X.Z. and C.L.; methodology, C.L.; validation, C.L., C.Z. and X.Z.; formal analysis, C.L. and X.Z.; investigation, C.L. and X.Z.; resources, X.Z.; data curation, C.L., C.Z. and W.Z.; writing—original draft preparation, C.L. and X.Z.; writing—review and editing, Q.F. and W.Z.; visualization, C.L. and C.Z.; supervision, Q.F. and X.Z.; funding acquisition, Q.F. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Major Science and Technology Project, grant number 2016ZX05025-001, and CNOOC major science and technology project, grant number KJGG2021-0501.

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: Author Chen Liu, Wensheng Zhou and Chi Zhang were employed by the State Key Laboratory of Offshore Oil and Gas Exploitation and CNOOC Research Institute Ltd. 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.

Abbreviations

The following abbreviations are used in this manuscript:

DB Davies–Bouldin

References

- Deng, R.; Dong, J.J.; Dang, L.X. Numerical simulation and evaluation of residual oil saturation in waterflooded reservoirs. *Fuel* **2025**, *384*, 134018. [CrossRef]
- Han, D.K. Discussions on concepts, countermeasures and technical routes for the redevelopment of high water-cut oilfields. *Pet. Explor. Dev.* **2010**, *37*, 583–591.
- Wang, D.; Liu, F.; Li, G.; He, S.; Song, K.; Zhang, J. Characterization and dynamic adjustment of the flow field during the late stage of waterflooding in strongly heterogeneous reservoirs. *Energies* **2023**, *16*, 831. [CrossRef]
- Bai, M.X.; Zhang, Z.C.; Liang, J.W. Identification and adjustment of streamline field in middle-high permeability sandstone oilfield. *Pet. Geol. Recovery Effic.* **2017**, *24*, 100–105.
- Chen, Y.S. *Reservoir Flow Field*; Petroleum Industry Press: Beijing, China, 1998.
- Liu, X.T.; Wang, P.Z.; Wang, Y.Z. Medium-high permeability, mature reservoir management study: An innovative methodology and case study. In Proceedings of the SPE International Oil and Gas Conference and Exhibition in China, Beijing, China, 8–10 June 2010.
- Chen, F.Z.; Jiang, H.Q.; Li, J.; Bai, X. Quantitative liquid field description method and its application. *J. Oil Gas Technol.* **2011**, *33*, 111–114+174.
- Ma, K.Q.; Li, A.; Guo, S.H.; Pang, J.Q.; Xue, Y.C.; Zhou, Z.H. Techniques for improving the water-flooding of oil fields during the high water-cut stage. *Oil Gas Sci. Technol.—Rev. IFP Energ. Nouv.* **2019**, *74*, 69. [CrossRef]
- Zhang, Q.L.; Jiang, R.Z.; Jiang, P.; Yao, Z.; Wang, Y.; Peng, W. Establishment and application of oil reservoir flow-field evaluating system. *Pet. Geol. Oilfield Dev. Daqing* **2014**, *33*, 86–89.
- Lu, F.M.; Wu, X.; Zhu, H.Y.; Zhang, Y.; Wang, R. Quantitative characterization method and application of reservoir flow field. *J. Southwest Pet. Univ. (Sci. Technol. Ed.)* **2020**, *42*, 111–120.
- Zhang, S.M.; Yang, Y. Study on the dynamic tracking model of flow field boundaries in water drive reservoirs at ultra-high water cut stage. *Pet. Geol. Recovery Effic.* **2021**, *28*, 91–99.
- Abdulkareem, M.A.; Martin, J.B. Streamline-based simulation of non-Newtonian polymer flooding. *SPE J.* **2010**, *15*, 895–905.
- Zhang, N.; Sun, J.; Ren, L.; Cao, J.; Chen, M. Streamline simulation: Recent trends in methodology development and engineering applications. *J. Xi'an Shiyou Univ. (Nat. Sci. Ed.)* **2021**, *36*, 32–41+57.
- Jia, H.; Deng, L.H. Oil reservoir water flooding flowing area identification based on the method of streamline clustering artificial intelligence. *Pet. Explor. Dev.* **2018**, *45*, 312–319. [CrossRef]
- Guo, Q.; He, S.M.; Meng, L.X. Study on gradational optimization of oil reservoir streamline field based on an artificial intelligence algorithm. *J. Pet. Explor. Prod. Technol.* **2019**, *9*, 1295–1306. [CrossRef]
- Zhao, P.Q.; He, S.M.; Cai, M.J.; Tao, Z.; Zhao, M.; Wu, X.; Ni, T.; Guo, Q.; Wei, P.; Wang, Q.; et al. Streamline simulation based vector flow field characterization and reconstruction method for high water cut reservoir. In Proceedings of the International Petroleum Technology Conference, Dhahran, Saudi Arabia, 13–15 January 2020.
- Li, S.S.; Feng, Q.H.; Zhang, X.M.; Liu, H.; Liu, L.; Huang, Y. Flow field characterization and evaluation method based on unsupervised machine learning. *J. Pet. Sci. Eng.* **2022**, *215*, 110599. [CrossRef]
- Liu, F.; Zhou, W.S.; Liu, N.X.; Li, K.; Zhang, K.; Cao, C.; Qin, G.; Cao, C.; Yang, R. Flow field description and simplification based on principal component analysis downscaling and clustering algorithms. *Front. Earth Sci.* **2022**, *9*, 804617. [CrossRef]
- Geng, Z.L.; Jiang, H.Q.; Chen, M.F.; Sun, M.R.; Zeng, Y.X. A new method for quantitative characterization on remaining oil potential in high water cut oil reservoirs. *Pet. Geol. Recovery Effic.* **2007**, *14*, 100–102+118.
- Yu, C.L. A prediction method of phase permeability curve reflecting the limit of water drive. *Spec. Oil Gas Reserv.* **2014**, *21*, 123–126+157.
- Liu, H.H.; Yan, Y.Q. New theory and practice of characterizing the phase permeability relationship in the ultra-high water-cut period. *J. Southwest Pet. Univ. (Nat. Sci. Ed.)* **2019**, *41*, 127–136.
- Li, S.S.; Feng, Q.H.; Zhang, X.M.; Yu, C.; Huang, Y. A new water flooding characteristic curve at ultra-high water cut stage. *J. Pet. Explor. Prod. Technol.* **2023**, *13*, 101–110. [CrossRef]
- Wang, J.Q.; Shi, C.F.; Ji, S.H.; Li, G.L.; Chen, Y.Q. A new type of water drive characteristic curve in the ultra-high water cut period. *Pet. Explor. Dev.* **2017**, *44*, 955–960. [CrossRef]
- Zhao, P.Q.; Hao, H.M.; Ni, T.L.; Li, H.G.; Tao, Z.Q.; Ma, Y.H. A novel technology: Reservoir seepage geophysics. *Oil Geophys. Prospect.* **2019**, *54*, 719–728.
- Yang, J.C.; Zhao, C. Survey on K-means clustering algorithm. *Comput. Eng. Appl.* **2019**, *55*, 7–14+63.
- Wang, S.; Liu, C.; Xing, S.J. Review on K-means clustering algorithm. *J. East China Jiaotong Univ.* **2022**, *39*, 119–126.
- Ikotun, A.M.; Ezugwu, A.E.; Abualigah, L.; Abuhaija, B.; Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Inf. Sci.* **2023**, *622*, 178–210. [CrossRef]
- Liu, Y.C.; Li, Z.M.; Xiong, H.; Gao, X.D.; Wu, J.J. Understanding of internal clustering validation measures. In Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia, 13–17 December 2010.

29. Liu, F.L.; Liu, S.Y. Dynamic Fuzzy Clustering Algorithm Based on Weighted Difference Degree. *J. Jilin Univ. Sci. Ed.* **2019**, *57*, 574–582.
30. Jansen, J.D.; Fonseca, R.M.; Kahrobaei, S.; Siraj, M.M.; Van Essen, G.M.; Van den Hof, P.M.J. The Egg model: A geological ensemble for reservoir simulation. *Geosci. Data J.* **2014**, *1*, 192–195. [CrossRef]
31. Li, Y.C.; Jia, D.L.; Wang, S.L.; Qu, R.Y.; Qiao, M.X.; Liu, H. Surrogate model for reservoir performance prediction with time-varying well control based on depth generative network. *Pet. Explor. Dev.* **2024**, *51*, 1287–1300. [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.

Article

Inversion of Gravity Anomalies Based on U-Net Network

Fei Yu ^{1,2}, Guiju Wu ^{1,2,*}, Yufei Xi ^{3,*}, Fan Wang ^{1,2}, Jiapei Wang ^{1,2}, Rui Zhang ^{1,2} and Qinghong Long ^{1,2}

¹ Institute of Seismology, China Earthquake Administration, Wuhan 430071, China; 13731310544@163.com (F.Y.); wangfan@hubdzj.gov.cn (F.W.); wangjiapei@hubdzj.gov.cn (J.W.); zr_18674828695@163.com (R.Z.); longqinhong0823@163.com (Q.L.)

² Hubei Earthquake Agency, Wuhan 430071, China

³ Institute of Hydrogeology and Environmental Geology, Chinese Academy of Geological Sciences, Shijiazhuang 050061, China

* Correspondence: wuguiju@hubdzj.gov.cn (G.W.); xiyufei@mail.cgs.gov.cn (Y.X.); Tel.: +86-15337192960 (G.W.); +86-13931136710 (Y.X.)

Abstract: The deep learning-based gravity anomaly inversion method addresses the complex challenge of deriving subsurface density variation models from surface gravity anomaly data. In order to generate various geological environments and their corresponding surface gravity anomaly datasets, three-dimensional density models considering different spatial locations and density variations are created in this paper. At the same time, the residual module and spatial attention mechanism are introduced into the U-Net architecture to improve the learning ability and inversion accuracy of complex geological structures. Experimental results demonstrate that the proposed method achieves the high-precision reconstruction of density variation models in complex anomaly environments, with a model residual error lower than 3%. Additionally, the inversion results of the density change and the gravity change in the Longshoushan fault zone show that the 2022 Menyuan MS6.9 earthquake is in the middle of the positive and negative density changes, which verifies the applicability of the U-Net network in the field of gravity change data, highlighting the method's value in the real-world environment.

Keywords: density variation; U-Net; gravity anomaly

1. Introduction

Gravity inversion, a classical non-unique inverse problem, has seen significant progress through the development of various regularization techniques aimed at improving its stability and accuracy. Minimum volume constraints and their refinements have effectively restricted the solution space, resulting in more compact and interpretable inversion models [1–7]. Similarly, inversion methods that incorporate minimum gradient support constraints have greatly enhanced the sharpness and compactness of reconstructed structures [8–10]. The integration of depth-weighting functions into objective functions has helped mitigate the “skin effect”, thereby improving the vertical resolution of subsurface models [11,12]. Furthermore, the incorporation of the velocity model has significantly improved the reconstructions of density models by aligning inversion results with plausible geological environments [13].

Despite these advancements, challenges remain due to the reliance on initial models, susceptibility to local minima, and limitations in computational efficiency and stability. These issues are particularly pronounced in complex geological structures, where constructing accurate initial models is both technically challenging and computationally intensive [14].

Overcoming these challenges requires innovative methodologies to enhance the applicability of gravity inversion in complex subsurface systems. The integration of big data technologies and artificial intelligence has led to substantial advancements in geophysical data processing. Machine learning techniques, such as clustering [15,16], decision trees [17], support vector machines [18], Bayesian methods [19,20], and artificial neural networks (ANNs) [21], have gradually replaced the traditional reliance on initial models and sensitivity matrices. Its powerful global optimization capability makes machine learning one of the key research directions in geophysics [22].

Deep learning (DL), a specialized subset of machine learning, has tackled ill-posed inverse problems through hierarchical feature extraction [23,24]. This approach eliminates the need for manual feature engineering while improving the approximation of complex functions [25]. Studies by Kim and Nakata have demonstrated deep learning's effectiveness in addressing intricate data challenges in large-scale inversion tasks [26]. In the geophysical data processing, deep learning algorithms have significantly improved the resolution and adaptability in reconstructing complex geological structures [27–29]. For instance, Wang et al. developed a novel method for magnetotelluric data inversion using genetic neural networks, significantly enhancing accuracy [30]. Pusyrev employed fully convolutional neural networks (FCNs) for the two-dimensional inversion of vertical magnetic dipole source data from boreholes [31]. Additionally, Pusyrev and Swidinsky applied deep learning to improve the precision and efficiency in the one-dimensional inversion of marine controlled-source electromagnetic data [32]. Noh et al. successfully integrated convolutional neural networks (CNNs) into airborne electromagnetic data inversion [33]. Li et al. utilized FCNs to reconstruct velocity models from seismic data, optimizing both resolution and efficiency [34]. Zhang et al. applied an optimized U-Net architecture to analyze gravity and gravity gradient data and achieved the accurate detection of density anomalies with improved efficiency [35].

The main challenge faced by deep learning in the inversion of gravity data is the difficulty in obtaining real underground spatial data, compounded by the potential for simulated data to deviate from real physical laws. To address this, this paper employs forward modeling to ensure that the results adhere to real physical laws. Additionally, deep learning models often exhibit a poor generalization to unseen data, necessitating transfer learning tailored to the specific requirements of different scenarios. Despite these challenges, the trained model demonstrates high efficiency, strong nonlinear mapping capabilities, and end-to-end automation.

This study employs a U-Net architecture to develop a nonlinear mapping model that directly links gravity anomaly inputs to density variation anomaly outputs. U-Net is a classical deep learning model, and its symmetry plays important roles, such as image segmentation and geophysical inversion. Two important mathematical properties—reflection symmetry and scale invariance—endow the model with an excellent robustness and generalization ability. Through supervised training and validation on a targeted dataset, this approach effectively inverts observed gravity data, realizes the time-varying gravity data inversion of three-dimensional density variation, completes accurate model reconstruction, and offers an advanced solution for exploring complex geological systems.

2. Dataset Construction

In this study, a sample dataset generator is developed to simulate three-dimensional structural anomalies and their corresponding gravity anomalies. This addresses the challenge of obtaining accurate and comprehensive real subsurface interface data. The generator can be configured with parameters that closely reflect subsurface conditions, enabling the creation of density variation anomalies with specific shapes, sizes, and spatial distributions,

while simultaneously modeling the resulting surface gravity anomalies. Specifically designed for training deep learning networks, this tool facilitates the generation of realistic and representative training datasets, thereby enhancing model performance. The dataset includes both density variation anomalies and forward-simulated surface gravity anomalies, ensuring the validity and reliability of the simulation process.

2.1. Construction of 3D Density Variation Anomaly Models

A sample dataset representing a density variation anomaly model in a subsurface half-space is constructed. This initiative aims to mitigate overfitting and address potential issues arising from insufficient datasets during training. The generated data were then used to compute surface gravity anomalies through numerical simulations, ultimately constructing a comprehensive gravity anomaly dataset.

Model Space Construction

The subterranean half-space is systematically divided into multiple uniform cuboidal modeling units, as shown in Figure 1. The influence of each individual unit on surface observation points is calculated sequentially. The surface gravity anomaly is the cumulative contribution of all the grid units in the subsurface half-space to the surface.

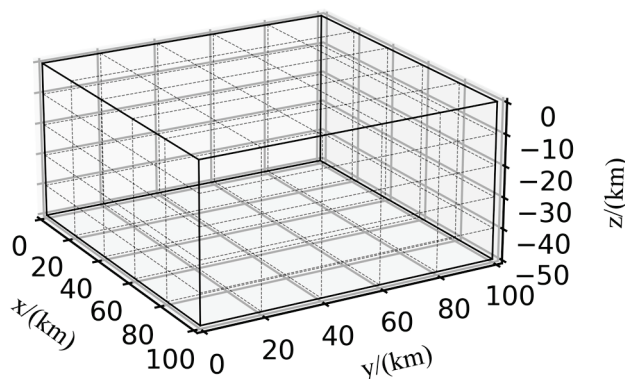


Figure 1. Three-dimensional half-space grid.

When selecting the location of the abnormal body in the model, the crust above the Moho surface where shallow earthquakes usually occur is selected, and 50 km is selected here.

Based on the seismogenic range associated with pre-earthquake gravity anomalies, a horizontal extent of 100 km is selected. After several calculations and considering the computational load, a grid size of 1 km is chosen to ensure model accuracy. The subsurface half-space selected for this study is $100 \text{ km} \times 100 \text{ km} \times 50 \text{ km}$, and the size of each grid is $1 \text{ km} \times 1 \text{ km} \times 1 \text{ km}$. This configuration is chosen to accurately simulate the density variation anomaly of the subsurface region where the earthquakes occurred.

2.2. Forward Modeling of 3D Models

2.2.1. Model Generation

In the subsurface half-space of $100 \text{ km} \times 100 \text{ km} \times 50 \text{ km}$, a total of 8100 single density variation anomalies are systematically generated and used as the foundational dataset for training deep learning models, as shown in Figure 2. Using a 4090 graphics card, an Intel 19-13900K CPU, and 128 GB of RAM are used to generate the model with MATLAB 2022. The total processing time is 7 min and 6 s. Each density variation anomaly is set as $16 \text{ km} \times 16 \text{ km} \times 8 \text{ km}$. These anomalies are spaced 5 km in x and y directions, and 4 km vertically in z direction. At each spatial location, 10 distinct density anomalies, ranging

from -0.5 g/cm^3 to $+0.5 \text{ g/cm}^3$, are generated, ensuring an even distribution of density variation anomalies in the entire subsurface half-space.

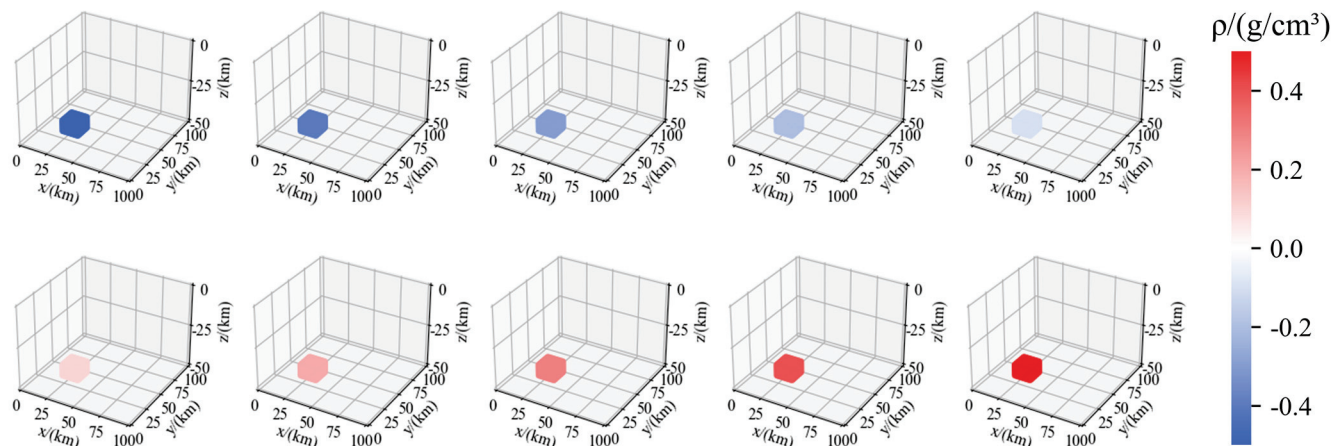


Figure 2. Generation of single density anomaly body. Anomalous body is located at (25, 40, 5) km, and density variation interval is 0.1 g/cm^3 .

2.2.2. Model Combination

In this study, a deep learning model training dataset is created. Standard models with varying numbers of anomalies are selected based on the characteristics of real gravity changes, rather than simply providing more shapes. This approach ensures that the model maintains a strong generalization ability and does not lose representativeness due to overfitting in real-world environments. A three-dimensional density variation anomaly model is constructed by integrating random selection with spatial constraints, the boundary test and random offset are used to ensure that the positions of anomalous bodies do not overlap and do not exceed the subsurface half-space, allowing for realistic and complex anomaly shape simulations. A diverse and intricate composite model is systematically developed by selecting multiple anomalies from a pre-generated independent outlier dataset and superimposing them onto a target composite grid. Using a 4090 graphics card, an Intel 19-13900K CPU, and 128 GB of RAM the single anomalous body model is combined into models with 2, 3, and 5 anomalous bodies, respectively. The total processing time is 1 h, 32 min, and 5 s.

Spatial consistency is maintained through the implementation of a placement algorithm that randomly selects offsets within the grid while verifying placement locations to prevent anomalies overlapping. If a conflict occurs, the algorithm dynamically recalibrates the positions until a valid configuration is achieved. Additionally, a filtering mechanism is employed to eliminate density variation anomalies that exceed the mesh size.

The generated anomaly bodies in the x, y, and z spaces exhibit varying characteristics including positive and negative values, orders of magnitude, spatial positions, and quantities. The anomalies include single-layer positive density anomalies at a depth of 10 km, as shown in Figure 3a; two density variation anomaly bodies at a depth of 40 km, as shown in Figure 3b; and different positive and negative density variation magnitudes, as shown in Figure 3c,d. Additionally, there are anomaly bodies located at depths of -20 km and -40 km , as well as more complex anomaly bodies characterized by simultaneous positive and negative densities, varying magnitudes, and deep differences in Figure 3d. These anomaly bodies are used as input for model training.

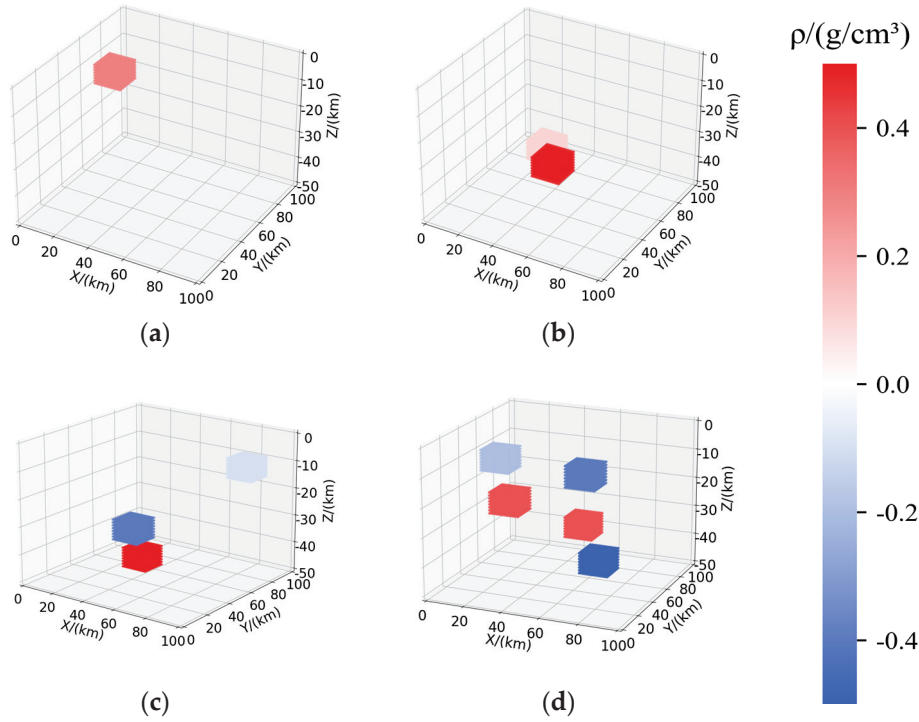


Figure 3. Density anomaly in subsurface assembly: (a) single density variation anomaly body; (b) two density variation anomaly bodies; (c) three density variation anomaly bodies; and (d) five density variation anomaly bodies.

Adopting a batch processing strategy can effectively mitigate memory overflow issues that arise when generating large volumes of data rapidly. This approach addresses challenges related to computational efficiency and memory management. Furthermore, it facilitates the generation of combination exceptions within a controlled time frame, thereby enhancing overall system performance. By dynamically adjusting batch sizes (MATLAB provides memory commands (Windows only) to monitor available memory; these commands are used to dynamically resize a batch, set a minimum memory threshold of 2 GB, and release memory when the threshold is reached) and utilizing garbage collection mechanisms (when the memory usage is too large to run, the batch size is actively reduced to run), the method significantly reduces memory consumption and enhances computational stability, facilitating the efficient processing of large datasets. These strategies ensure the efficient handling of large datasets while maintaining computational stability.

The resulting composite anomaly model is stored in .npz format for seamless integration into subsequent analysis pipelines. This approach not only provides a comprehensive and diverse dataset for training and evaluating deep learning models but also establishes an advanced framework for constructing complex, high-fidelity data, advancing anomaly detection and geophysical interpretation.

2.2.3. Data Forward Modeling

The theoretical formulation for the forward computation of gravity anomalies at the observation point $P(x, y, z)$, corresponding to a single cuboid model element, is presented as follows:

$$g = G \sum_x \sum_y \sum_z \frac{\rho V_z}{r^3} \quad (1)$$

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \quad (2)$$

where r represents the distance between the observation point $P(x,y,z)$ and the grid center of gravity (x_0,y_0,z_0) , g is the surface gravity value, and G is the universal gravitation constant, which is approximately $6.67430 \times 10^{-11} \text{ m}^3/\text{kg}/\text{s}^2$. P is the density of the anomaly body, V is the volume of the anomaly body, and z is the depth perpendicular to the surface.

Figure 4 shows the gravitational effects of one, two, three, and five underground anomaly bodies as calculated using Equation (1), corresponding to Figure 3. We have established a vertical range for the anomaly bodies that extends 50 km below the surface in order to ensure the applicability of density variation anomalies within the real-space Moho surface and above. Using a 4090 graphics card, an Intel 19-13900K CPU, and 128 GB of RAM accelerates parallel computing using CUDA in python. The total processing time is 11 h, 5 min and 3 s.

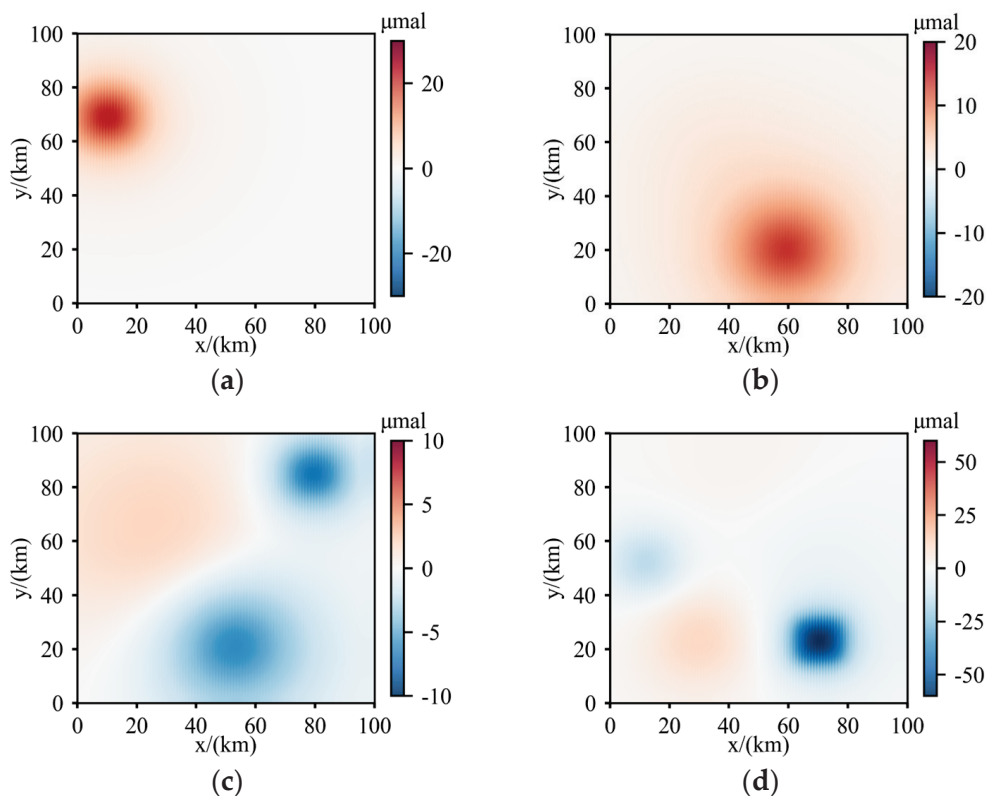


Figure 4. The surface forward modeling data: (a) the surface gravity effect of a single positive anomaly body; (b) the surface gravity effect of two positive anomaly bodies with a density difference; (c) the surface gravity effect of three positive/negative anomaly bodies; and (d) the surface gravity effect of five distinct density variation anomaly bodies.

When multiple anomaly bodies are present simultaneously with significant depth disparities, the surface gravity effect of a variation anomaly body characterized by low density variation may be obscured by the weight force effect from high-value density variation anomalies at the surface. This phenomenon, illustrated in Figure 4b, occurs when the distance between two positive density anomalies is small and the magnitude difference is significant. In this case, surface gravity anomalies generated by smaller anomalies can be masked by larger ones, which poses considerable challenges for future inversion work related to our model.

3. Inversion of Network Architecture

Compared to traditional gravity inversion methods, deep learning offers a direct approach to learning the complex nonlinear mapping between gravity anomalies and

subsurface density distribution through end-to-end training. This approach eliminates the need for explicit mathematical modeling or prior assumptions. In this paper, a neural network structure based on U-Net is constructed, which adopts coding–decoding architecture to hierarchically extract and reconstruct underground density features. By training on a large dataset of high-precision theoretical models, the network can automatically learn density distribution patterns under various geological structures and effectively adapt to complex underground environments.

3.1. The Main Architecture of U-Net

Long et al. introduced the Fully Convolutional Network (FCN) method by transforming the fully connected layers of traditional Convolutional Neural Networks (CNNs) into convolutional layers, thus enabling end-to-end learning [36]. This innovation significantly reduces the number of hyper-parameters associated with nodes, enhancing computational efficiency. Furthermore, it mitigates the data loss typically linked to fully connected layers, preserving critical information necessary for optimizing output predictions and enabling accurate and efficient dense pixel prediction.

Building upon U-Net, a well-known convolutional neural network architecture, this study introduces a novel and streamlined network design that optimizes both computational efficiency and model performance. Specifically, the proposed architecture reduces the overall depth of the network and minimizes the number of convolutional channels, thereby decreasing memory consumption and accelerating training while maintaining high accuracy in the label classification at the output stage.

A key innovation of the architecture is the removal of traditional skip connections, which helps speed up the network training process. However, this enhancement comes at the expense of some fine-grained details, as the attention mechanism prioritizes dominant spatial features over local variations. Despite this trade-off, the overall performance of the model benefits from the improved feature representation, resulting in more robust and stable inversion results.

This study integrates a carefully designed residual module that compensates for the absence of skip connections by facilitating an effective gradient flow. This modification directly addresses the gradient vanishing issues that often hinder the training of deep networks, thereby accelerating convergence and improving stability during optimization.

Moreover, we introduce a spatial attention mechanism to further refine feature extraction. Unlike conventional convolutional layers that process all spatial features uniformly, the spatial attention module adaptively highlights important regions, enabling the network to focus on more relevant spatial structures in gravity field inversion. By doing so, the model can better capture large-scale patterns in the input data, which is crucial for improving the accuracy of spatial dimension reconstruction.

Additionally, the inclusion of residual modules enhances the network's nonlinear approximation capabilities, allowing for a more precise reconstruction of spatial dimensions. As demonstrated in Figure 5, the proposed network outperforms conventional U-Net-based architectures, offering a more effective and efficient solution for geophysical applications.

The network is designed to train as much data as possible within a computing resource-constrained environment. For complex underground structural scenes, training focuses more on capturing the shapes and spatial locations of anomaly bodies rather than getting bogged down in intricate details. The incorporation of residual structures effectively addresses overfitting issues prevalent in small datasets, enhances model generalization capabilities, and produces smoother results.

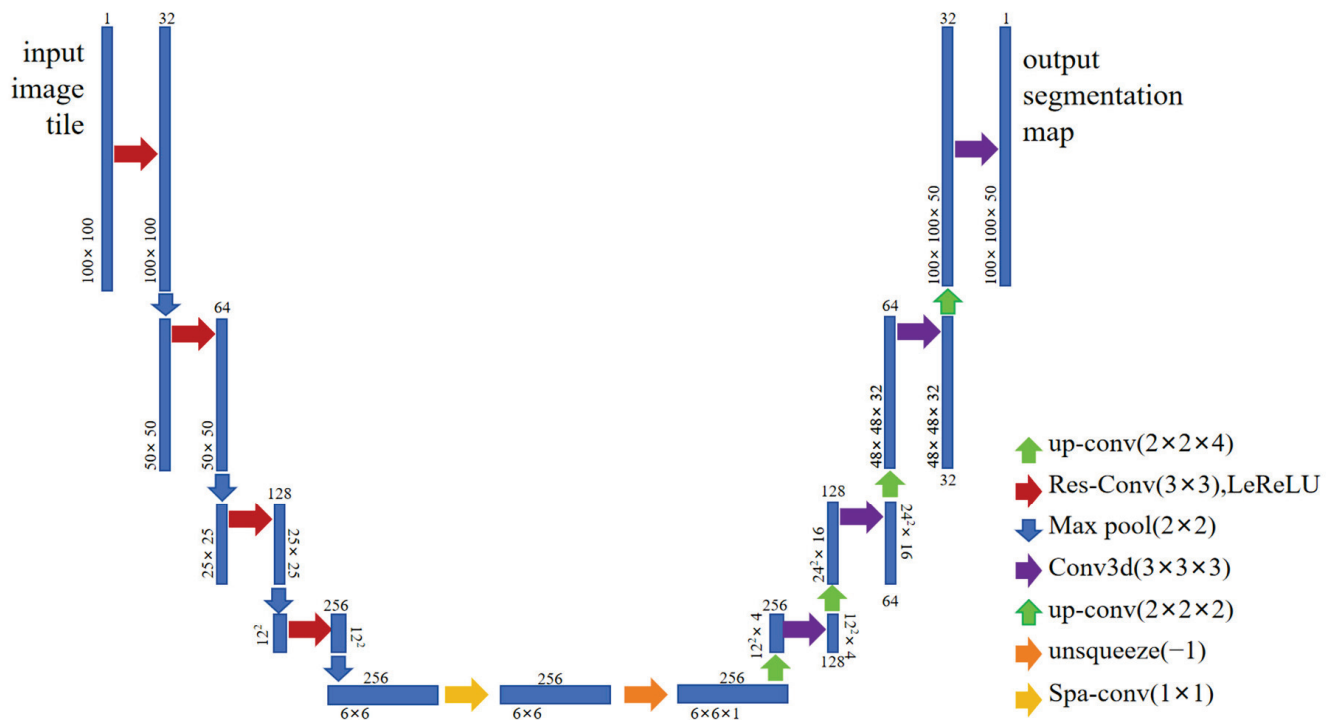


Figure 5. U-net network.

The Conv2D layer in the shortcut path is used to resize the input layer to a different dimension; in order to achieve the role of the interlayer transfer, the dimension is still unified. Notably, the Conv2D layer within this shortcut path does not incorporate any nonlinear activation functions, ensuring that the dimensions align correctly when adding the values from the shortcut residual block back to the main pathway (similar to the effect achieved by matrix multiplication with WSW_SWS). This layer applies a linear transformation to reduce the size of the input, ensuring compatibility with the subsequent addition steps, as shown in Figure 6 [37].

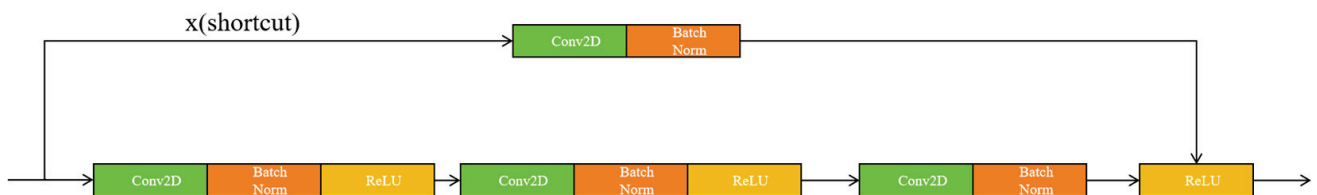


Figure 6. Residual module.

Ultimately, this design aims to enhance training stability and efficiency by mitigating issues related to vanishing gradients. It seeks to improve training performance and streamline optimization processes for deep networks. Furthermore, it ensures that the characteristics of underground anomalies are sufficiently minimized for effective extraction.

Placing the spatial attention mechanism between the encoder and decoder offers distinct advantages. This design leverages high-dimensional semantic features extracted by the encoder to further optimize feature distribution through spatial attention mechanisms prior to entering the decoder [38]. This approach enhances feature representation in critical regions while mitigating interference from irrelevant areas. Given that the encoder has already captured rich contextual information, introducing a spatial attention mechanism at this stage allows for a more precise provision of sensitive features related to the target region, thereby facilitating detail recovery during the decoding phase. In the context of

high-precision tasks, such as the 3D reconstruction addressed in this paper, implementing an attention mechanism before model dimensionality can significantly enhance model performance, preserve structural compactness, and maximize local feature retention while ensuring computational efficiency, as shown in Figure 7.

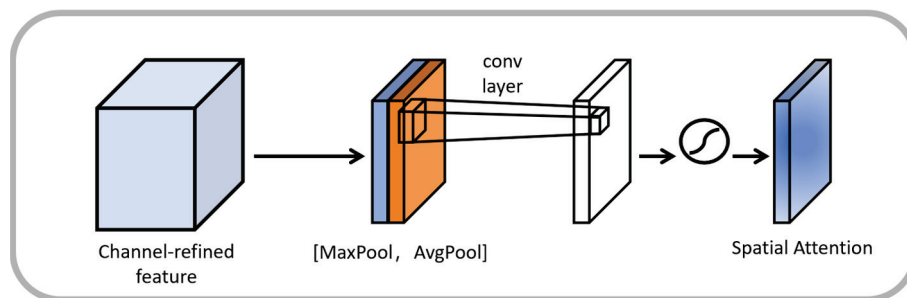


Figure 7. Spatial attention mechanism.

3.2. Regularization

In deep learning, the generalization ability of a U-Net network pertains to its predictive performance on previously unseen data [39,40]. Typically, datasets are divided into training and test sets, with network generalization assessed via test error metrics. To improve U-Net's generalization performance, two key factors must be considered. First, minimizing loss function values; and second, reducing discrepancies between training errors and test errors. An excessively high loss function may result in under-fitting, a condition that can be rectified by adjusting either the U-Net architecture or its training algorithm. Conversely, a significant gap between the training error and test error indicates overfitting within the model. For environments involving ample data availability, regularization methods present an effective solution.

Dropout is a widely employed regularization technique in the training of U-Net models. It randomly deactivates a subset of neuronal connections during each training iteration, primarily within the encoders and decoders of the U-Net architecture. This method mitigates the model's reliance on the co-adaptation between specific nodes, thereby simplifying the model structure and effectively reducing the risk of overfitting [41]. Additionally, Batch Normalization is frequently employed between convolutional layers in U-Net architectures. This regularization method serves to stabilize network performance and accelerate model convergence throughout the training process.

The study employs Dropout and Batch Normalization as regularization techniques, which are widely recognized for enhancing the generalization capability of neural networks. This enhancement improves the model's adaptability to diverse datasets. Consequently, this approach ensures that our experiments can more accurately process complex surface gravity data in real geological environments, ultimately leading to improved inversion results for subsurface density variation anomaly bodies.

3.3. Loss Function

Mean Square Error (MSE) is employed as the loss function to align with the inherent continuity of underground spatial representations. This selection ensures smooth and continuous reconstruction by minimizing differences between predicted and true values. In this implementation, MSE functions as the primary loss metric for quantifying discrepancies between predicted 3D data (i.e., model output) and target 3D data. Specifically, $MSELoss$ is computed as follows:

$$MSELoss = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3)$$

where y_i represents the actual target value (ground truth), and $\hat{y}_i$ denotes the target 3D data in this code. It signifies the predictive output of the model. N indicates the number of data samples.

This study introduces regional root-mean-square errors to address the disproportionate impact of high-error regions in traditional Root Mean Square Error (RMSE) assessments. The proposed method enhances the accuracy of the loss function within local contexts by independently evaluating the performance across sub-regions or subsets. It effectively mitigates the effects of spatial randomness in the distribution of subsurface anomaly bodies, where variations in size and location can result in significant errors in certain areas while causing only minor deviations elsewhere.

$$\text{RMSE}_{i,j,k} = \frac{1}{B_H \times B_W \times B_D} \sum_{x=0}^{B_H-1} \sum_{y=0}^{B_W-1} \sum_{z=0}^{B_D-1} \left(P_{i,j,k}(x, y, z) - T_{i,j,k}(x, y, z) \right)^2 \quad (4)$$

where B represents the unit block of the RMSE calculation, x, y, z represents the length, width. And depth of each block, $P_{i,j,k}$ represents the target value of the i, j, k block, and $T_{i,j,k}$ represents the predicted output value of the i, j, k block model.

3.4. Algorithm Selection

This study adopts the Adaptive Moment Estimation (Adam) algorithm to enhance the training process of the 3D underground density reconstruction network. Traditional Stochastic Gradient Descent (SGD) with a globally uniform learning rate frequently encounters challenges [42]. Given that each variable's dependency on the loss function varies, inappropriate learning rate settings can lead to a slow convergence for some variables and oscillations for others. To address this issue, the AdaGrad algorithm adjusts learning rates individually for each variable, allowing for more precise tuning. However, AdaGrad has a limitation: the rapid accumulation of squared gradients results in a progressively decreasing learning rate over time, which hinders the model's ability to reach an optimal solution.

RMSProp improves upon AdaGrad by using an exponentially decaying average of squared gradients, thereby enabling continuous training and preventing the algorithm from getting stuck in a local minimum. Building on RMSProp, Adam further incorporates momentum through the calculation of first-order and second-order moment estimates of the gradients [43]. This combination provides more robust update directions and helps stabilize the training process. Thanks to its momentum-enhancing features, Adam not only stabilizes training but also accelerates the optimization process, making it particularly effective for large datasets requiring hyper-parameter tuning. With its high computational efficiency and fast convergence, Adam is especially well-suited for complex 3D network training tasks.

Simultaneously, the ReduceLROnPlateau method is employed as a dynamic learning rate adjustment strategy. This approach monitors specific indicators, such as the losses on validation sets, to determine if they have plateaued during the model training process. If these monitored metrics do not show significant improvement over a defined period, the ReduceLROnPlateau mechanism automatically decreases the learning rate, thereby facilitating the more stable convergence of our three-dimensional inversion model.

3.5. Network Configuration

In this study, the activation function employed is the widely adopted Rectified Linear Unit (ReLU), as presented in Equation (5). Additionally, an enhanced variant proposed by Xu et al. is utilized [44]. ReLU is favored for its capability to induce sparsity and

alleviate the vanishing gradient problem, ensuring rapid convergence and minimizing computational cost.

$$f_{\text{ReLU}}(t) = \max(t, 0) \quad (5)$$

$$f_{\text{leakyReLU}}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \quad (6)$$

For the output layer, the Leaky ReLU function (depicted in Equation (6)) is implemented. This activation function effectively mitigates the dying ReLU issue, facilitating smooth gradient propagation and enhancing flexibility in capturing nonlinear features, making it particularly suitable for three-dimensional reconstruction tasks.

To extract features, this study adopts maximum pooling as the pooling strategy. Compared with average pooling, maximum pooling more effectively highlights salient features [45], which is essential for retaining critical information during feature extraction. This method significantly contributes to the construction of underground three-dimensional anomaly weights.

4. Inversion and Discussions

4.1. Single Anomaly Body Inversion

During each training epoch, the input data were processed by the U-net architecture, which is based on convolutional neural networks (CNNs). The input images and target body data were loaded in batches, with each batch containing 32 samples. RMSE losses were computed for smaller blocks consisting of $10 \times 10 \times 5$ voxels. This block-by-block comparison with the target data allowed for the quantification of the difference between the predicted output and the target using the RMSE. This approach reduced computational requirements while maintaining the accuracy of local data.

4.1.1. Classification Results

A total of 50 epochs were conducted throughout the training process. After each epoch, training losses, validation losses, and the current learning rate were calculated and documented. Initially, a modest learning rate of 10^{-5} was established alongside the ReduceLROnPlateau learning rate scheduler; this approach entails halving the learning rate when no further reduction in validation loss occurs. This strategy aims to enhance training stability while mitigating overfitting risks. Additionally, an early stopping criterion was implemented during the training process: if there was no significant improvement in the validation loss for five consecutive epochs, training would be terminated prematurely to prevent overfitting.

In the training of a single anomaly body, the utilization of the Adam optimizer markedly accelerates convergence, mitigates fluctuations during training, and substantially reduces the overall convergence time, as shown in Figure 8a. The model exhibits commendable characteristics throughout the training process: it achieves rapid convergence within the first five epochs with a significant reduction in the loss function. As training progresses, by the 10th epoch, the rate of loss declines, plateaus, and the model stabilizes. By the 30th epoch, the training process is essentially complete, with successful convergence. Ultimately, the residual error is maintained below 2%, which underscores the model's superior fitting capability and provides robust assurance for subsequent applications, as shown in Figure 8b.

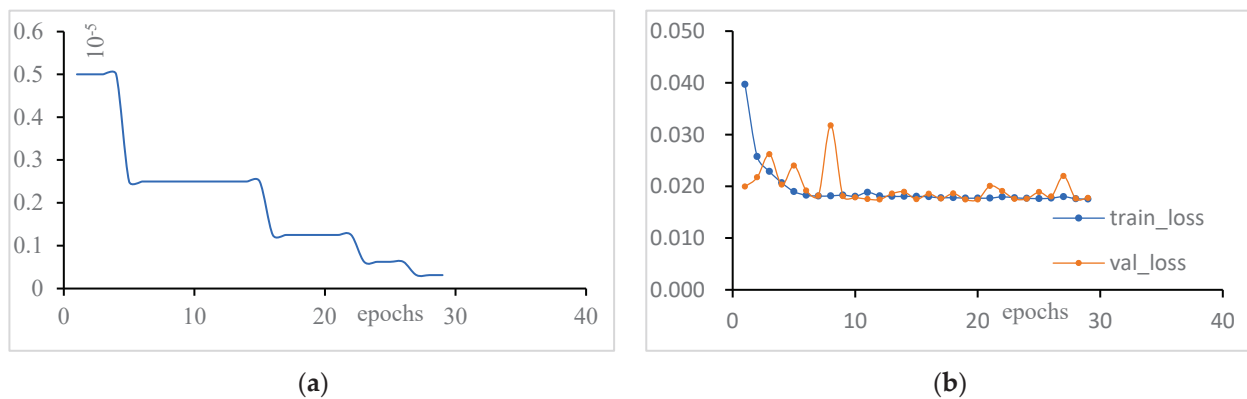


Figure 8. Single model network training: (a) single model training learning rate and (b) single model network training loss.

4.1.2. Forecast Results

Under the inversion framework based on deep learning, the results obtained from inverting a single anomaly body demonstrate remarkable effectiveness. In comparison to traditional inversion methods, deep learning-based approaches exhibit significant advantages. Deep learning models possess the capability to learn complex nonlinear mapping relationships from extensive training datasets, which enhances their flexibility when addressing anomaly bodies situated within heterogeneous or intricate backgrounds.

Particularly in environments involving a solitary anomaly body, as shown in Figure 9, these models can accurately capture the density distribution characteristics of the anomaly by optimizing the loss function and even detect subtle variations. On the horizontal slice, there is no diffusion at all, the model size is good, and there is some upward diffusion on the z direction, but the overall shape and size are intact. This proficiency allows the model not only to reconstruct the overall structure of the anomaly body but also to maintain a high level of precision regarding its finer details.

The model demonstrates a commendable performance in both the recovery of form and the spatial position reconstruction of the anomaly body. It presents a clear and complete geometric structure, with calculated values closely approximating real values and exhibiting minimal errors. This outcome validates that the model can accurately invert anomalous bodies under known conditions while maintaining high stability and accuracy, irrespective of whether the density of these bodies is positive or negative. The reconstruction results for both positive and negative density anomalies indicate that the model effectively captures the density distribution characteristics associated with different types of anomaly bodies, thereby circumventing the reliance on specific density intervals typical of traditional methods. In contrast to conventional gravity inversion techniques, deep learning models do not depend on prior simplified assumptions; instead, they adaptively learn complex feature distributions related to both positive and negative anomalies through training. This inherent flexibility not only enhances their efficacy in detecting individual positive and negative anomaly bodies but also lays a robust foundation for addressing intricate geological challenges involving abnormal bodies with diverse properties in future research. Overall, the deep learning-based inversion model exhibits significant flexibility and adaptability. It efficiently identifies both the nature and spatial distribution of singular abnormal entities while simultaneously offering theoretical possibilities for subsequent inversions involving multi-density structures as well as actual surface gravity data.

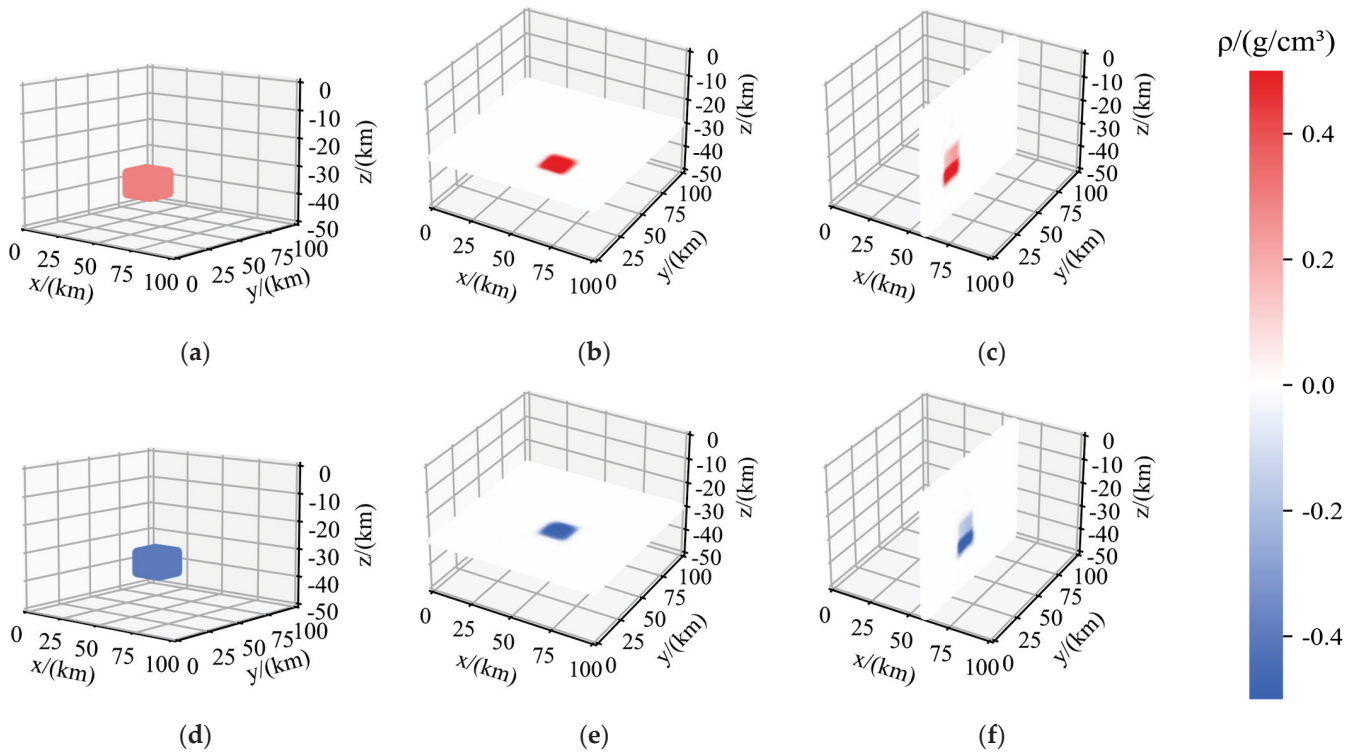


Figure 9. Inversion results: (a) single positive density variation anomaly body, anomalous bodies exhibiting density variation of 0.5 g/cm^3 , and center situated at $(60, 60, -32) \text{ km}$; (b) horizontal slice of inversion results in (a), and slice position is $z = -30 \text{ km}$; (c) vertical slice of inversion result in (a), and slice position is at $x = 60 \text{ km}$; (d) single negative density variation anomaly body, and anomalous bodies exhibiting density variation of -0.5 g/cm^3 and center situated at $(60, 60, -32) \text{ km}$; (e) horizontal slice of inversion results in (d), and slice position is at $z = -30 \text{ km}$; (f) vertical slice of inversion result in (d), and slice position is $x = 60 \text{ km}$.

4.2. Inversion of Multiple Classified Anomaly Bodies

In this study, datasets containing varying numbers of anomalies (1, 2, 3, and 5) were combined to generate a total of 32,000 forward and inverse models with different anomaly counts (8000 of each type). The approach was employed to evaluate the robustness of the network across diverse model configurations and to assess the performance of complex models. The block RMSE loss function was utilized during network training. The primary objective was to enable the model to accurately predict anomalies within three-dimensional data.

During each training epoch, the input data were processed by the U-Net architecture, which is based on convolutional neural networks (CNNs). The input images and target body data were loaded in batches, with each batch containing 96 samples. RMSE losses were computed for smaller blocks consisting of $10 \times 10 \times 5$ voxels. This block-by-block comparison with the target data allowed for the quantification of the difference between the predicted output and the target using the RMSE. This research seeks to enhance the predictive accuracy for anomalies present in three-dimensional datasets while managing computational efficiency through localized error calculations. The inversion of the standard model for complex subsurface anomaly bodies was achieved.

4.2.1. Classification Result

A total of 50 epochs were conducted during the multiple anomaly bodies training process. After each epoch, the training losses, validation losses, and current learning rate were calculated and recorded. Initially, a small learning rate of 10^{-5} was set, and

the ReduceLROnPlateau learning rate scheduler was employed. When the validation loss ceased to decrease, the learning rate was halved. This strategy aimed to enhance training stability and mitigate overfitting. Additionally, an early stopping criterion was implemented during the training process: if there was no significant improvement in the validation loss for five consecutive epochs, training would be terminated prematurely to prevent overfitting.

During the training process, the Adam optimizer significantly enhances the convergence rate of the model, minimizes fluctuations during training, and substantially reduces the time required for convergence, as shown in Figure 10a. The model exhibits favorable training characteristics: it achieves rapid convergence within the first 10 epochs with a marked decrease in the loss function. As training progresses, the rate of decline in the loss function stabilizes, and the model reaches stability by the 30th epoch. By the 40th epoch, the training process is essentially complete, and the model successfully converges. The final residual error of the model is maintained below 5%, demonstrating its superior fitting performance and establishing a robust foundation for subsequent applications, as shown in Figure 10b.

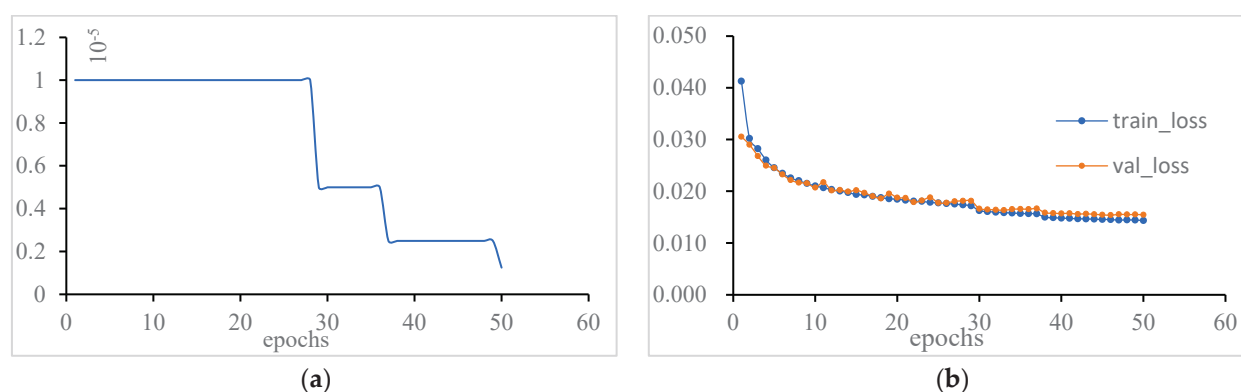


Figure 10. Combinatorial model network training: (a) combinatorial model training learning rate and (b) combinatorial model network training loss.

4.2.2. Forecast Results

1. Two anomalies

A two anomaly body characterized by significant differences in burial depth is employed to evaluate the model. This design aims to simulate the features of geological anomalies at varying depths, thereby allowing for a comprehensive assessment of the model's adaptability in addressing complex variations in burial depth. The combination of high and low buried depth differences effectively verifies the sensitivity and inversion accuracy of the model, providing a solid foundation for the further optimization of the algorithm.

The inversion results of the model are commendable, as shown in Figure 11. As it accurately restores the primary geometric features, the two anomalous bodies have no diffusion, size, and range inversion effects in the horizontal direction and the z direction is good. Regarding positional accuracy, the inversion model effectively captures both the horizontal and vertical positions of these anomalies, with depth errors maintained within a minimal range. This demonstrates the model's superiority in spatial analytical capabilities.

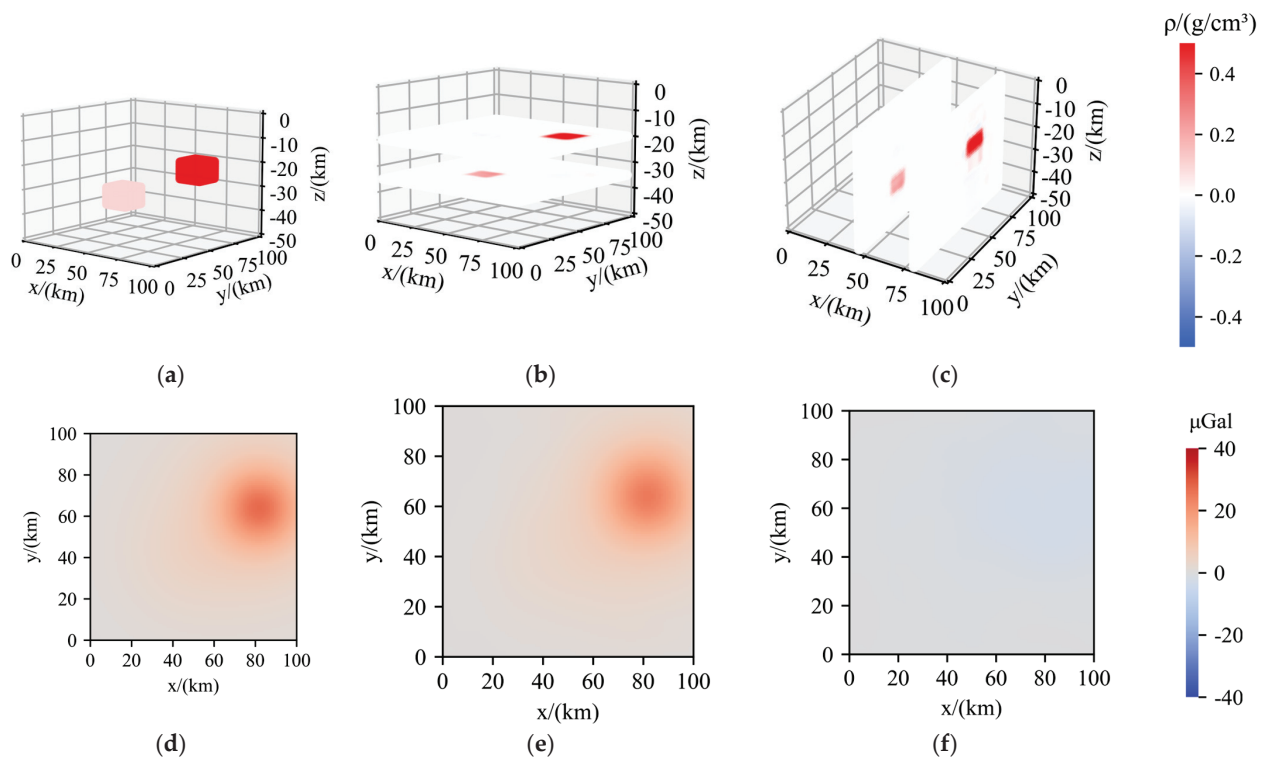


Figure 11. Inversion results of two anomaly bodies: (a) Two positive density variation anomaly bodies, characterized by density variations of 0.2 g/cm^3 and 0.5 g/cm^3 , respectively. First body is centered at coordinates $(45, 45, -35) \text{ km}$ and second body is centered at $(80, 80, -20) \text{ km}$; (b) horizontal slices of inversion results in (a) and slices' positions are at $z = -35 \text{ km}$ and $z = -20 \text{ km}$; (c) vertical slices of inversion result in (a), and slices' positions are at $x = 45 \text{ km}$ and $x = 80 \text{ km}$; (d) density (a) forward surface data; (e) inversion results of surface gravity data; and (f) residuals of (d,e).

In terms of density distribution characteristics, the inversion results provide detailed insights into the internal structure and boundary changes in both anomalies. The transition between density values for these anomalies and their surrounding background is remarkably smooth. Furthermore, the characterization of boundaries for each anomaly body is distinctly clear, indicating that the input data have been well restored by the model.

It is noteworthy that not only do these inversion results accurately recover both the shape and relative positioning of the two anomaly bodies, they also reveal potential patterns regarding their underground interaction characteristics. For instance, a gradual change in relative density gradients between these two bodies suggests that the model can sensitively detect anomaly signal characteristics even within complex environments. Additionally, a satisfactory resolution at depth ensures the preservation of fine structures within both anomaly bodies in a vertical context—providing an effective reference for multi-layer geological body inversions in intricate settings.

In the process of verifying the inversion results, it was observed that the residuals remain stable within $\pm 5 \mu\text{Gal}$ when comparing ground gravity observations with values generated by the inversion. Furthermore, the distribution of these residuals is uniform, exhibiting no significant systematic bias or concentrated areas of error. This strongly indicates that the model's inversion results possess a robust capability to accurately represent the ground gravity signal, demonstrating a high degree of consistency between observation and simulation data. Such findings further validate both the physical binding force and stability inherent in the model. Additionally, the smooth distribution observed in the residual region suggests that the inversion process effectively mitigates the unnecessary amplification of noise, thereby minimizing its impact on the result accuracy.

2. Three anomaly bodies

When testing complex model tests involving three anomaly bodies, a scenario was designed that incorporated both positive and negative anomalies while preserving the depth variations among them. The design not only increased the data diversity but also introduced more challenging targets, thereby substantially enhancing the complexity of the inversion process. This approach was intended to comprehensively evaluate the model's adaptability to complex geological structures.

The model exhibits a strong capacity for reverse signal capture, as shown in Figure 12. The abnormal body with density variations of -0.2 g/cm^3 and $+0.2 \text{ g/cm}^3$ in the shallow layer has no diffusion and deformation in its contour boundary and its value in the horizontal direction and depth direction. The abnormal body with a density variation of $+0.3 \text{ g/cm}^3$ in the deep layer has a certain diffusion boundary deformation ambiguity due to the influence of the abnormal body with a relatively close horizontal distance in the shallow layer, but the overall shape and value are correct. The model accurately restores the characteristics of both positive and negative anomaly bodies while effectively representing the main geometric features, depth distribution, and density variation anomalies within the subsurface model over time. Regarding positional accuracy, the inversion model successfully captures both horizontal and vertical positions of the three anomalies; furthermore, the depth errors are maintained within a minimal range—highlighting the superiority of this model in spatial analytical capabilities.

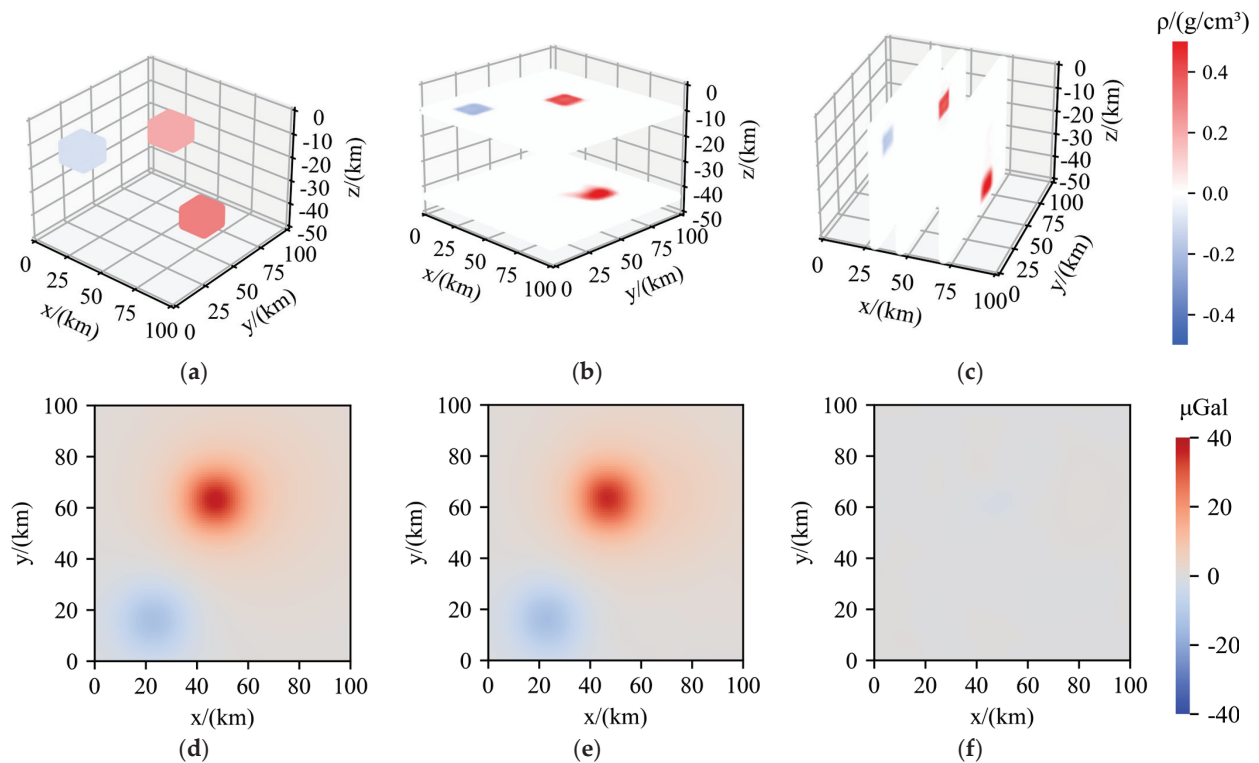


Figure 12. Inversion results of three anomaly bodies: (a) Three positive density variation anomaly bodies, characterized by density variations of -0.2 g/cm^3 , 0.2 g/cm^3 , and 0.5 g/cm^3 , respectively. First body is centered at coordinates 24, 24, -10 km , and second body is centered at 50, 60, -12 km . Third body is centered at 64, 64, -42 km ; (b) horizontal slices of inversion results in (a), and slices' positions are at $z = 10 \text{ km}$ and $z = -42 \text{ km}$; (c) vertical slices of inversion result in (a), and slices' positions are at $x = 30 \text{ km}$, $x = 48 \text{ km}$, and $x = 70 \text{ km}$; (d) density (a) forward surface data; (e) inversion results of surface gravity data; and (f) residuals of (d,e).

Concerning the density distribution characteristics, the inversion results provide detailed insights into the internal structures and boundary changes associated with the three anomalies. The transition between the density values among these anomalies and their surrounding background is smooth without any abrupt shifts or abnormal high/low density readings. Additionally, while there is a minor overflow observed locally at certain points the overall shape stability is preserved, thus ensuring that input data models are restored effectively.

In the process of verifying the inversion results, the residuals remain stable within $\pm 2 \mu\text{Gal}$ when comparing ground gravity observations with the values generated by the inversion. The distribution is uniform, exhibiting no significant systematic bias or concentrated areas of error. The model demonstrates an enhanced performance in environments where both positive and negative density signals are present, resulting in a surface residual that is lower than that observed for the two anomaly bodies.

The random introduction of three outliers adds complexity to the verification process by incorporating both positive and negative values along with deep differences. This effectively showcases the model's superior capability in handling complex environments. The findings indicate that the model not only sensitively captures subtle features within intricate scenes but also maintains an excellent performance amidst variations in depth and noise suppression. These attributes establish a robust foundation for the extensive application of this model in practical geological exploration, further validating its effectiveness and applicability for multi-objective and multi-level geological body inversion tasks.

3. Five anomaly bodies

In the complex density variation model test involving five anomaly bodies, a carefully designed scenario was established in which both positive and negative anomaly bodies coexisted, thereby comprehensively assessing the model's adaptability to intricate and realistic geological structures. This design not only enhances data diversity but also presents a more challenging target for analysis. Particularly, when confronted with both positive and negative anomalies alongside varying signal gradients within the region, the complexity of inversion is significantly heightened.

The test results indicate that the model's recognition capability remains exceptional even in complex situations, as shown in Figure 13. From -5 km near the surface to -40 km below the deep layer, the lowest density variation in the anomalous body is -0.5 g/cm^3 , and the largest density variation is 0.4 g/cm^3 , and the density variation is not as significant as -0.1 g/cm^3 , which achieves the inversion effect. The shape boundary of the deepest anomalous body is fuzzy, but the shape integrity is still accurate. For positive anomalies, the density distribution effectively highlights high-value regions, while for negative anomalies, the inversion results accurately restore both the depth and extent of low-value areas. This demonstrates that the model can adeptly manage differences between positive and negative densities while maintaining high accuracy and preventing diffusion or the blurring of edge values. The precise inversion of both positive and negative density characteristics further substantiates its performance potential when dealing with multi-source heterogeneous signals in intricate geological contexts.

Moreover, it is noteworthy that the model exhibits a commendable resolution in vertically inverted high and low anomaly bodies. Particularly, within environments involving deep structural anomalies, variations in high and low-density gradients are meticulously restored, with corresponding transition zones exhibiting a remarkable degree of continuity. This level of detail underscores the model's ability to sensitively capture minor structural changes along the vertical axis while providing an extensive characterization across multiple geological layers.

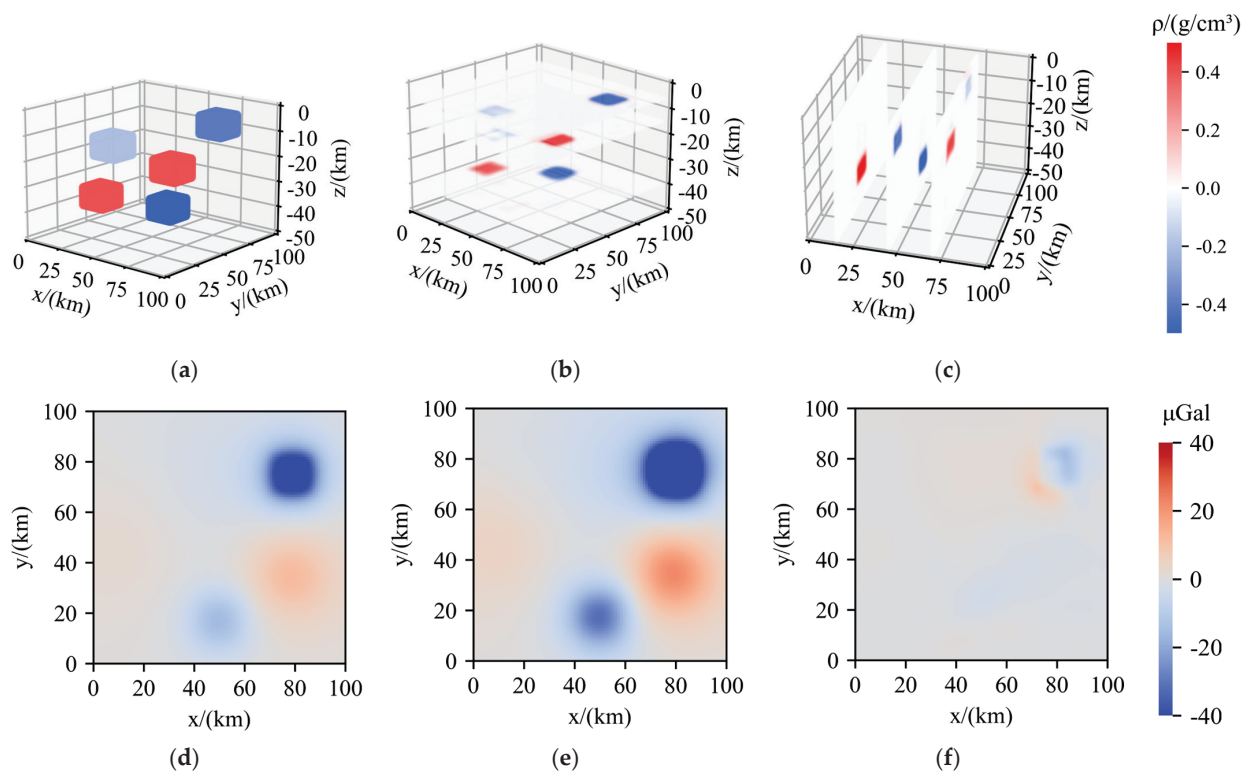


Figure 13. Inversion results of five anomaly bodies: (a) three positive density variation anomaly bodies, characterized by density variations of 0.4 g/cm^3 , 0.4 g/cm^3 , -0.1 g/cm^3 , -0.3 g/cm^3 , and -0.5 g/cm^3 , respectively. First body is centered at coordinates 12, 48, -40 km , second body is centered at 78, 32, -18 km , third body is centered at 50, 16, -10 km ; fourth body is centered at 80, 78, -16 km ; fifth body is centered at 48, 64, -42 km ; (b) horizontal slices of inversion results in (a), and slices' positions are at $z = -4 \text{ km}$, $z = -15 \text{ km}$, and $z = -40 \text{ km}$; (c) vertical slices of inversion result in (a), and slices' positions are at $x = 20 \text{ km}$, $x = 48 \text{ km}$, and $x = 72 \text{ km}$; (d) density (a) forward surface data; (e) nversion results of surface gravity data; and (f) residuals of (d,e).

A performance worthy of in-depth analysis is observed in the intersection region of the anomaly bodies characterized by drastic density changes. In this context, the model demonstrates remarkable resilience against signal mixing, effectively delineating the boundaries of each independent anomaly body with a high degree of stability. Despite these complex conditions, the model maintains an impressive level of attention to local details, indicating that its structural design is adept at managing various regional disturbances. Moreover, the model exhibits both stability and adaptability across different testing environments, including environments involving random arrangements of positive and negative anomalies as well as unique distributions characterized by varying elevations. This flexibility enables it to address inversion requirements in straightforward situations while consistently delivering reliable results under more intricate conditions. Additionally, these experiments unveil potential expansion capabilities within the model framework, thereby supporting future practical applications in multi-domain geological inversion.

In summary, this study validated the model through randomly generated five anomaly bodies featuring unequal positives and negatives alongside alternating high and low levels. The inversion outcomes reaffirmed both the adaptability and accuracy inherent in this methodology; it exhibited exceptional sensitivity in feature detection while maintaining stability and precision when addressing complex geological structures. Collectively, these test results establish a robust foundation for practical applications in geological inversion and further substantiate the efficacy of this approach when confronted with challenging environments.

5. Model Application and Verification

The gravity measurement data around the Longshoushan fault zone, collected from 2020 to 2021 by the Gravity Branch of the China Earthquake Administration, were subjected to inversion analysis, by integrating these data with the tectonic dynamic background of the northeastern margin of the Qinghai–Tibet Plateau [46]. We conducted a comprehensive analysis of density fluctuations and subsurface attributes preceding seismic events, thereby substantiating the practical applicability and intrinsic value of our model. According to information from the USGS, the Menyuan M_s 6.9 earthquake in 2022 occurred in a region characterized by long-term, left-lateral, and strike-slip shear and vertical uplift, with a focal depth of 15 km. The inversion results indicated significant density variation anomalies at various depths within the study area. The observation area spans $100\text{ km} \times 100\text{ km}$, while the surface gravity data interpolation is conducted at a resolution of $1\text{ km} \times 1\text{ km}$, aligning with the input parameters for our model. The subsurface was divided into cuboid units measuring $100\text{ km} \times 100\text{ km} \times 50\text{ km}$, each unit being sized at $1\text{ km} \times 1\text{ km} \times 1\text{ km}$. High-precision inversion research was conducted utilizing the U-Net network framework. The distribution pattern of residual errors within experimental data indicates that the peak values for surface gravity residual errors reach up to $15\text{ }\mu\text{Gal}$ but progressively decrease from peripheral areas towards the center of the investigation. In proximity to the epicenter, residual errors fall below $5\text{ }\mu\text{Gal}$ —well within acceptable limits—and exert minimal influence on the interpretations regarding underground structures in this region. This observation underscores that our model possesses strong regional focusing capabilities which effectively diminish secondary interference while minimizing impacts from external density variation anomalies on local inversions within our study area. However, in practical applications, the influence of density anomalies outside the region introduce interferences, leading to a poor spatial edge effect in the trained model and a slight reduction in the overall accuracy. Meanwhile, the limitations of single gravity inversion, such as poor shape, also exist in deep learning inversion. Overall, this network demonstrates high accuracy and adaptability when detecting large-scale anomalies as well as complex subterranean features.

Based on the available data, the U-Net network employed in this study was utilized to conduct the inversion analysis of the target region. A significant positive density anomaly of 0.2 g/cm^3 was identified at depths of 20–30 km northwest of the Tolai Mountain fault zone. In contrast, a negative density anomaly of 0.3 g/cm^3 was detected at depths ranging from 15 to 25 km southeast of the Lenglongling fault zone. These anomalies are horizontally distributed and parallel along their respective fault zones, with stress effects aligning with the orientations of these faults. This alignment exhibits a strong consistency with the left-lateral strike-slip characteristics observed by the USGS during the Menyuan M_s 6.9 earthquake event at a focal depth of 15 km. Figure 14d illustrates the distribution of 20 km underground sections, faults, and earthquake aftershocks in the epicentral area. The distribution of earthquake aftershocks was consistent with the inversion results. By relying on the spatial relationship between fault zones and density changes, the U-Net-based inversion results effectively explain pre-earthquake gravity variations. Additionally, the distribution of density changes correlates with the strike of the Lenglongling fault zone, showing an increase on the northwest side and a decrease on the southeast side of the epicenter. These findings align with the focal mechanism solution and epicenter shift reported by Dai et al., as well as the spatial extent of major aftershock distributions [47]. Furthermore, regional gravity anomalies reveal fluctuating characteristics in the Moho surface, indicating substantial changes at approximately 50 km of depth. The results demonstrate that the enhanced U-Net network successfully captures complex gravity signals while significantly reducing secondary interference, showcasing robust generalization capabilities across diverse geological contexts. This provides reliable

technical support and valuable tools for investigating density variations and intricate subsurface structures.

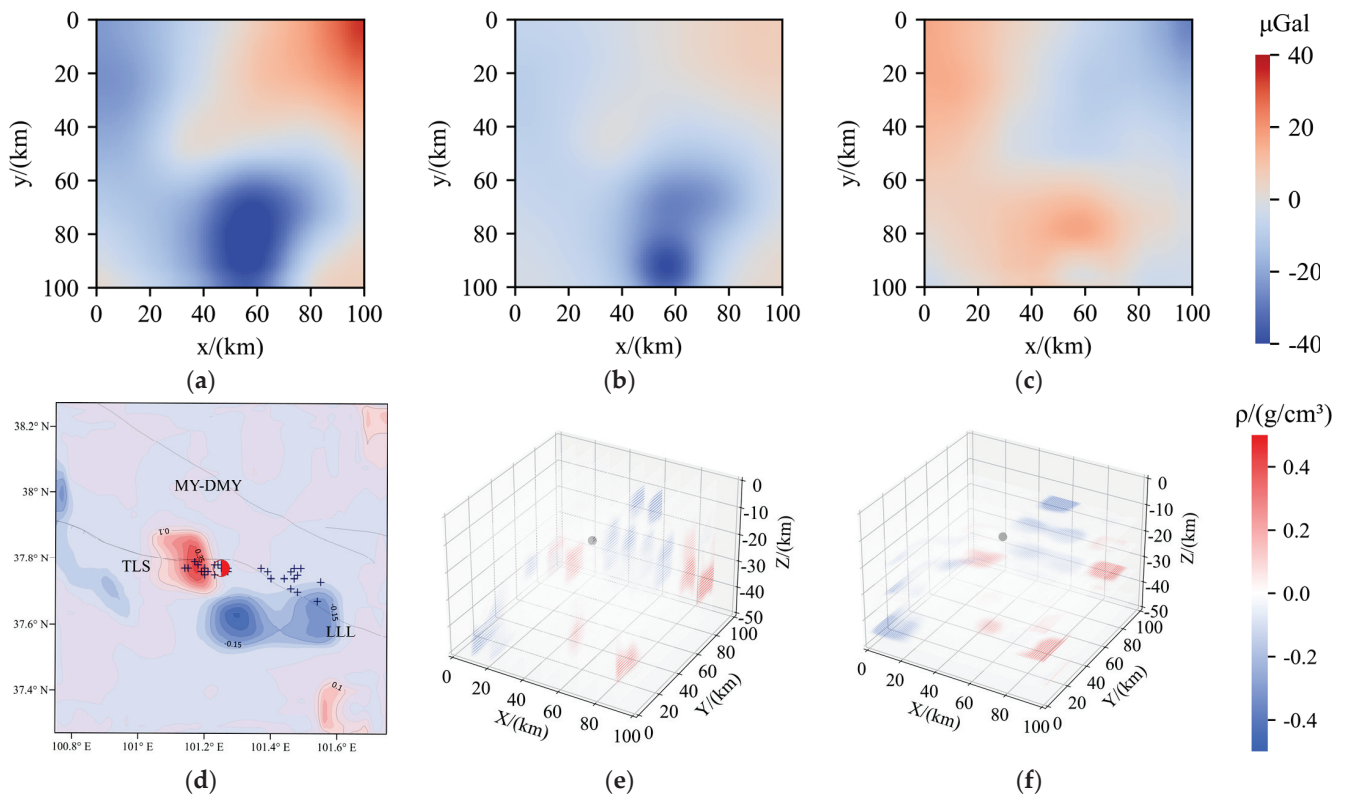


Figure 14. Gravity anomaly and underground structure: (a) measured gravity anomalies within 100 km of gate source, $\times$ is the aftershock and the red hemisphere is the main shock; (b) inversion results of measured gravity anomalies within 100 km of gate source; (c) residuals of (a,b); (d) earthquake with respect to fault location and underground 20 km density changes abnormally, and black cross is Menyuan aftershock of M_{5.6.9} earthquake, red star represents M_{5.6.9} earthquake, TLS is Tuolaishan fracture, MY-DMY is Minle-Damaying fracture, LLL is Lenglongling fracture; (e) vertical slice of inversion result in (a); and (f) horizontal slices of inversion results in (a), and black dot is Menyuan M_{5.6.9} earthquake.

In summary, this study demonstrates the feasibility of applying the U-Net-based gravity anomaly inversion approach in real-world geological investigations. The improved network effectively captures complex gravitational signatures, mitigates secondary influences, and effectively generalizes across diverse subsurface environments, providing robust support for practical geological challenges.

6. Conclusions

In this study, the U-Net network model is employed for the inversion of underground anomaly bodies. Compared with traditional seismic inversion methods, this model can automatically extract features from a broad range of gravity anomaly data, thereby eliminating the reliance on manual feature extraction and prior assumptions inherent in conventional approaches. This capability significantly enhances both the efficiency and flexibility of the inversion process, making it particularly well suited for complex geological environments.

1. During the theoretical model testing phase, a RMSE loss function based on partition calculations is introduced. This method takes into account the spatial distribution characteristics of underground anomaly bodies and effectively optimizes inversion accuracy.

2. In model tests, not only are the shape, size, and boundaries of subsurface anomalies accurately constrained, but also the residual gravity from the surface flow is maintained below 5 μGal . These results further validate both the feasibility and superiority of using U-Net networks for gravity anomaly inversion.
3. In practical applications, U-Net networks have successfully addressed inversion challenges in geological contexts, as demonstrated by their ability to identify underground density variation changes preceding the Menyuan M_s 6.9 earthquake. The focal depth matches the one produced by the USGS while keeping surface gravity residuals within 15 μGal . Additionally, the distribution and focal mechanisms of aftershocks are consistent with those obtained using other researchers' GNSS methods, thereby demonstrating U-Net's capability in addressing non-unique gravity inversion issues.

Whether in theoretical testing or real-world earthquake case studies, U-Net networks have proven to be accurate and robust for gravity anomaly inversion, offering innovative methodologies for exploring density changes in complex geological environments.

Author Contributions: Methodology, and writing—original draft, F.Y.; data curation and writing—review and editing, project administration, and funding acquisition, G.W. and Y.X.; resources, and data curation, F.W. and J.W.; data curation, R.Z. and Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (NSFC) under grant (No 42074172, No 42204089).

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 the confidentiality requirements of gravity actual measurements.

Conflicts of Interest: Authors Fei Yu, Guiju Wu, Fan Wang, Jiawei Wang, Rui Zhang, and Qinghong Long were employed by the Hubei Earthquake Agency. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Last, B.J.; Kubik, K. Compact gravity inversion. *Geophysics* **1983**, *48*, 713–721. [CrossRef]
2. Guillen, A.; Menichetti, V. Gravity and magnetic inversion with minimization of a specific functional. *Geophysics* **1984**, *49*, 1354–1360. [CrossRef]
3. Barbosa, V.C.F.; Silva, J.B.C. Generalized compact gravity inversion. *Geophysics* **1994**, *59*, 57–68. [CrossRef]
4. Li, Y.G.; Oldenburg, D.W. 3-D inversion of gravity data. *Geophysics* **1998**, *63*, 109–119. [CrossRef]
5. Silva, J.B.C.; Barbosa, V.C.F. Interactive gravity inversion. *Geophysics* **2006**, *71*, 696–699. [CrossRef]
6. Dias, F.J.S.S.; Barbosa, V.C.F.; Silva, J.B.C. 3D gravity inversion through an adaptive-learning procedure. *Geophysics* **2009**, *74*, I9–I21. [CrossRef]
7. Bertete-Aguirre, H.; Cherkaev, E.; Oristaglio, M. Non-smooth gravity problem with total variation penalization functional. *Geophys. J. Int.* **2002**, *149*, 499–507.
8. Portniaguine, O.; Zhdanov, M.S. Focusing geophysical inversion images. *Geophysics* **1999**, *64*, 874–887. [CrossRef]
9. Zhdanov, M.S.; Robert, E.; Souvik, M. Three-dimensional regularized focusing inversion of gravity gradient tensor component data. *Geophysics* **2004**, *69*, 925–937. [CrossRef]
10. Zhdanov, M.S. New advances in regularized inversion of gravity and electromagnetic data. *Geophys. Prospect.* **2009**, *57*, 463–478. [CrossRef]
11. Chasseriau, P.; Chouteau, M. 3D gravity inversion using a model of parameter covariance. *J. Appl. Geophys.* **2003**, *52*, 59–74. [CrossRef]
12. Commer, M. Three-dimensional gravity modelling and focusing inversion using rectangular meshes. *Geophys. Prospect.* **2011**, *59*, 966–979. [CrossRef]
13. Xu, R.; Wu, G.; Wang, G.; Fu, G.; Yu, F.; Wang, X.; Liu, L. Density structure of the upper and middle crust in the northeast Dabie Orogen revealed by terrestrial gravity surveys. *Geol. J.* **2023**, *59*, 137–154. [CrossRef]

14. Xu, S.; Yu, H.; Li, H.; Gang, T.; Yu, J. The inversion of apparent density based on the separation and continuation of the potential field. *Chin. J. Geophys.* **2009**, *52*, 1592–1598. (In Chinese) [CrossRef]
15. Sun, J.J.; Li, Y.G. Multidomain petrophysically constrained inversion and geology differentiation using guided fuzzy c-means clustering. *Geophysics* **2015**, *80*, ID1–ID18. [CrossRef]
16. Liu, S.; Jin, S.G. 3-D gravity anomaly inversion based on improved guided fuzzy C-Means clustering algorithm. *Pure Appl. Geophys.* **2020**, *117*, 1005–1027. [CrossRef]
17. Barak, S.; Abedi, M.; Bahroudi, A. A knowledge-guided fuzzy inference approach for integrating geophysics, geochemistry, and geology data in a deposit-scale porphyry copper targeting, Saveh, Iran. *Boll. Geofis. Teor. Appl.* **2020**, *61*, 159–176. [CrossRef]
18. Li, M.; Li, Y.; Wu, N.; Tian, Y.; Wang, T. Desert seismic random noise reduction framework based on improved PSO-SVM. *Acta Geod. Geophys.* **2020**, *55*, 101–117. [CrossRef]
19. Guo, R.W.; Dosso, S.E.; Liu, J.X.; Dettmer, J.; Tong, X. Non-linearity in Bayesian 1-D magnetotelluric inversion. *Geophys. J. Int.* **2011**, *185*, 663–675. [CrossRef]
20. Jiang, F.B.; Dai, Q.W.; Dong, L. Nonlinear inversion of electrical resistivity imaging using pruning Bayesian neural networks. *Appl. Geophys.* **2016**, *13*, 267–278. [CrossRef]
21. Al-Garni, M.A. Interpretation of some magnetic bodies using neural networks inversion. *Arab. J. Geosci.* **2009**, *2*, 175–184. [CrossRef]
22. Zhang, Q.; Zhou, Y.Z. Big data will lead to a profound revolution in the field of geological science. *Geol. Sci.* **2017**, *52*, 637–648. (In Chinese) [CrossRef]
23. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
24. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–445. [CrossRef] [PubMed]
25. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
26. Kim, Y.; Nakata, N. Geophysical inversion versus machine learning in inverse problems. *Lead. Edge.* **2018**, *37*, 894–901. [CrossRef]
27. Wang, H.; Yan, J.; Fu, G.; Wang, X. Current status and application prospect of deep learning in geophysics. *Prog. Geophys.* **2020**, *35*, 0642–0655. (In Chinese) [CrossRef]
28. Jin, K.H.; McCann, M.T.; Froustey, E.; Unser, M. Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.* **2017**, *26*, 4509–4522. [CrossRef]
29. Liu, B.; Guo, Q.; Li, S.C.; Liu, B.C.; Ren, Y.X.; Pang, Y.H. Deep learning inversion of electrical resistivity data. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5715–5728. [CrossRef]
30. Wang, H.; Liu, M.; Xi, Z.; Peng, X. Magnetotelluric inversion based on BP neural network optimized by genetic algorithm. *Chin. J. Geophys.* **2018**, *61*, 1563–1575. (In Chinese) [CrossRef]
31. Puzyrev, V. Deep learning electromagnetic inversion with convolutional neural networks. *Geophys. J. Int.* **2019**, *218*, 817–832. [CrossRef]
32. Puzyrev, V.; Swidinsky, A. Inversion of 1D frequency- and time-domain electromagnetic data with convolutional neural networks. *Comput. Geosci.* **2020**, *29*, 104681. [CrossRef]
33. Noh, K.; Yoon, D.; Byun, J. Imaging subsurface resistivity structure from airborne electromagnetic induction data using deep neural network. *Explor. Geophys.* **2020**, *51*, 214–220. [CrossRef]
34. Li, S.C.; Liu, B.; Ren, Y.X.; Chen, Y.K.; Yang, S.L.; Wang, Y.H.; Jiang, P. Deep-learning inversion of seismic data. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 2135–2149. [CrossRef]
35. Zhang, Z.; Liao, X.; Cao, Y.; Hou, Z. Joint Gravity and Gravity Gradient Inversion Based on Deep Learning. *Chin. J. Geophys.* **2021**, *64*, 1435–1452. [CrossRef]
36. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015. [CrossRef]
37. 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), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
38. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. [CrossRef]
39. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding Deep Learning Requires Rethinking Generalization. In Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
40. Chan, B.; Deng, W.; Du, J. Noisy softmax: Improving the generalization ability of DCNN via postponing the early softmax saturation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
41. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. *arXiv* **2012**, arXiv:1207.0580.

- 42. Hinton, G.E.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
- 43. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
- 44. Xu, B.; Wang, N.Y.; Chen, T.Q.; Li, M. Empirical evaluation of rectified activations in convolutional networks. *arXiv* **2015**, arXiv:1505.00853.
- 45. Boureau, Y.L.; Ponce, J.; LeCun, Y. A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 111–118.
- 46. Jin, X.; Wang, S.; Jiang, X.; Zhang, L. Coseismic deformation and slip distribution of the M_W 6.9 Menyuan, Qinghai earthquake revealed by Sentinel-1A SAR imagery. *Prog. Geophys.* **2022**, *37*, 2267–2274. (In Chinese) [CrossRef]
- 47. Dai, D.Q.; Yang, Z.G.; Sun, L. Rupture Process of the M_S 6.9 Menyuan, Qinghai, Earthquake on January 8, 2022. *Acta Seismol. Sin.* **2023**, *45*, 814–822. [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.

Article

Research and Application of ROM Based on Res-PINNs Neural Network in Fluid System

Yuhao Liu ^{1,*}, Junjie Hou ¹, Ping Wei ², Jie Jin ¹ and Renjie Zhang ¹¹ China Aerospace Academy of Systems Science and Engineering, Beijing 100089, China² China Academy of Space Technology, Beijing 100089, China

* Correspondence: 13930491501@139.com

Abstract: In the design of fluid systems, rapid iteration and simulation verification are essential, and reduced-order modeling techniques can significantly improve computational efficiency and accuracy. However, traditional Physics-Informed Neural Networks (PINNs) often face challenges such as vanishing or exploding gradients when learning flow field characteristics, limiting their ability to capture complex fluid dynamics. This study presents an enhanced reduced-order model (ROM): Physics-Informed Neural Networks based on Residual Networks (Res-PINNs). By integrating a Residual Network (ResNet) module into the PINN architecture, the proposed model improves training stability while preserving physical constraints. Additionally, the model's ability to capture and learn flow field states is further enhanced by the design of a symmetric parallel neural network structure. To evaluate the effectiveness of the Res-PINNs model, two classic fluid dynamics problems—flow around a cylinder and Vortex-Induced Vibration (VIV)—were selected for comparative testing. The results demonstrate that the Res-PINNs model not only reconstructs flow field states with higher accuracy but also effectively addresses limitations of traditional PINN methods, such as vanishing gradients, exploding gradients, and insufficient learning capacity. Compared to existing approaches, the proposed Res-PINNs provide a more stable and efficient solution for deep learning-based reduced-order modeling in fluid system design.

Keywords: fluid system; Res-PINNs; flow around a cylinder; vortex-induced vibration

1. Introduction

Modeling and simulation (M&S) of fluid systems involve using computational techniques to represent the varying states and key behavioral characteristics of fluids. By solving system model equations through computer simulations, detailed information such as velocity fields, pressure distributions, and temperature distributions can be obtained. This approach enables iterative validation, performance testing, and identification of potential issues in the design system without the need for physical experiments, thereby enhancing safety and reliability. However, in computational simulations, spatial flow field regions are typically discretized using grids. Increasing the number of grids significantly raises computational costs. Moreover, due to the complexity of fluid dynamics equations—especially in cases involving complex fluid behaviors like multiphase flows and turbulence—more computational resources are required, making it challenging to achieve rapid solutions for fluid problems [1,2]. Consequently, ROM methods are necessary to accelerate simulation speed and reduce the complexity of solving fluid dynamics problems.

ROM is a well-established method for reducing computational time, proposed several years ago [3]. ROM simplifies the original system into a lower-order system that

retains key behaviors and characteristics, thereby decreasing computational resources and increasing efficiency while accurately capturing the system's dynamic behavior. Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD) are the most conventional ROM methods that help to lower computational costs while maintaining sufficient accuracy [4] and have been widely applied in related fields. In recent years, with the significant improvements in computational power, Artificial Intelligence (AI) techniques such as deep learning (DL) have been extensively applied to surrogate modeling and ROM [5]. Deep-learning methods effectively reduce the dimensionality of system models, reconstructing high-dimensional model characteristics into lower-dimensional features, thus enabling more efficient system simulation and analysis. These methods are now widely used across multiple engineering fields, such as aerospace [6], materials engineering [7], and structural engineering [8], demonstrating substantial advantages. For example, Halder [9] successfully used a Recurrent Neural Network (RNN) model to predict the force at the next time step based on the current aileron motion state in a wing-gust and aeroelastic interaction system, proving the effectiveness of DL in dynamic system modeling. Similarly, Zhang [10] utilized a Deep Neural Network (DNN) to predict force variations in a vortex-induced vibration system given fluid parameters and boundary conditions, further validating the potential of DL in complex fluid dynamics problems. Additionally, Li [11] employed a Long Short-Term Memory (LSTM) network to capture the dynamic characteristics of airfoil pitching and plunging systems in transonic flow across multiple Mach numbers, enabling rapid acquisition of force and motion responses, and demonstrating its value in various flow states. Despite the significant progress achieved by DL methods in surrogate modeling for fluid dynamics and structural engineering, these approaches typically reconstruct only partial information about the flow field state. For more complex simulation problems, such as fluid control or fluid–structure interactions, it is necessary to fully reconstruct the entire flow field state, as it represents essential parameters for the system's response. Furthermore, due to the inherently “black-box” nature of neural networks, which lack explicit constraints based on physical laws, there are limitations in objectively describing physical phenomena.

To enhance the interpretability of DL models, Raissi et al. [12] proposed a PINN framework that embeds Partial Differential Equations (PDEs) or other differential equations directly into the neural network. PINNs integrate physical equations into the neural network, ensuring that the model learns not only from data but also remains consistent with the underlying physical laws, driven by both data and physical equations [13]. When simulating fluid system problems, the governing equations of fluid dynamics can be embedded into the neural network as prior knowledge. The model can then utilize spatiotemporal discrete data of flow field states and structural movements to reconstruct the changes in flow fields and structural responses [14]. Currently, substantial research has been conducted worldwide using PINN models to address fluid dynamics problems. For instance, Bararnia [15] extended the application of PINN to solve problems involving viscous and thermal boundary layers. They selected three benchmark problems for thermal boundary layers and explored the impact of equation nonlinearity and unbounded boundary conditions on adjusting the network's width and depth to obtain reasonable solutions. Tang et al. [16] proposed a transfer learning-enhanced PINN that reconstructs flow field and structural information using varying numbers of training sets. They also introduced a step-wise iterative training strategy to train the PINN model, effectively reducing the model's dependence on large datasets and lowering training costs. Similarly, Arzani et al. [17] combined mathematical equations of blood flow with measurement data, using PINN to reconstruct near-wall blood flow and wall shear stress in arterial flow. However, when applying PINN to ROM, gradient vanishing or exploding problems frequently occur. These

issues often arise because the physical constraints—such as PDEs—embedded into the loss function during network training can introduce diversity and complex nonlinearities. This causes the gradient of the loss function to exhibit extremely large or small variations in certain regions. Additionally, when dealing with high-dimensional physical problems, the network may need to learn complex nonlinear mappings. The characteristics of these high-dimensional problems make neural networks prone to falling into “flat regions” or “saddle points”, leading to vanishing gradients, or encountering regions with extreme variations, resulting in sudden large increases in gradients. Consequently, the network fails to accurately learn from the data, leading to unsatisfactory results. Moreover, this increases both the training time and computational costs.

To address the aforementioned challenges, this study proposes PINNs based on residual modules, named Res-PINNs. The approach begins by designing a parallel neural network grounded in the concept of symmetry, which transmits sparse spatiotemporal state information of the flow field—divided across different dimensions—into the network. This improves both the accuracy and stability of the model when tackling complex problems. Additionally, residual neural network modules are incorporated into the model to mitigate gradient vanishing or explosion issues during training, thereby enhancing the reconstruction accuracy of high-dimensional flow field states and enabling more precise capture of the flow dynamics. To assess the effectiveness of the proposed model, it was applied to classical fluid dynamics problems, including flow around a cylinder and VIV. The results demonstrate that the model not only alleviates the gradient vanishing and explosion problems commonly encountered in traditional PINN training, but also achieves a more accurate flow field reconstruction. This substantially improves the model’s overall performance and generalization capability. The proposed approach enriches ROM methods and provides a fresh perspective on fluid system modeling and simulation. In comparison to existing research, this study makes three key contributions: (1) Embedding ResNet modules into the PINN framework: The study introduces ResNet modules into the PINN framework to address gradient vanishing and explosion issues during training, thereby enhancing the model’s stability and reliability. (2) Utilizing a parallel neural network architecture: This study adopts a symmetric parallel neural network architecture, which comprehensively learns data across multiple dimensions, improving the accuracy of the Res-PINNs model’s reconstruction and reducing errors. (3) Application to classical fluid dynamics problems: The Res-PINNs model is applied to classical fluid dynamics problems, such as flow around a cylinder and vortex-induced vibration, in the development of ROM for various scenarios. This application not only validates the model’s applicability and effectiveness but also provides an innovative solution for spacecraft system design, simulation validation, and iterative optimization, with significant practical implications.

2. Materials and Methods

This section will provide a detailed overview of the fundamental concepts related to the PINN neural network model, as well as the proposed model and its improvements in this study.

2.1. Neural Network Model

Neural network models, with their powerful nonlinear mapping and feature extraction capabilities, can approximate any bounded continuous function to arbitrary precision, as stated by the universal approximation theorem. This capability reduces the complexity of the original function, providing an effective approach for establishing ROM of systems. Among these models, the Fully Connected Neural Network (FCNN) is the most classical structure. An FCNN primarily consists of an input layer, an output layer, and one or

more hidden layers. The neurons in these layers, along with their interconnections, form a directed graph representing the neural network:

$$z_l = f_l(W_l^T z_{l-1} + b_l) \quad (1)$$

In this context, Z_l represents the l -th hidden layer, while W_l and b_l denote the weight matrix and bias vector for the l -th layer, respectively. The function f_l represents the nonlinear activation function, with common choices including the Rectified Linear Unit (ReLU) function, the sigmoid function, and others. The activation function's purpose is to transform a linear equation into a nonlinear one, thereby enhancing the expressiveness of the neural network and improving its ability to handle nonlinear problems. Neural network models require training only on the weight and bias parameters, which reduces computational cost while maintaining the accuracy of the results, thus providing an approximate solution for complex problems.

When solving problems using a neural network, a black-box model is typically constructed to map the input-output relationship. The solving process can be represented as follows:

$$U(x, t, \theta) \approx \dot{U}(x, t, \theta) = z_l(x, t, \theta, W, b) \quad (2)$$

Here, θ represents the parameters in the equation that need to be trained, $U(x, t, \theta)$ denotes the exact or true outcome, and $\dot{U}(x, t, \theta)$ represents the result computed by the neural network. This result is obtained by minimizing the loss function, which can be expressed as follows:

$$\begin{aligned} L(W, b) &= \frac{1}{N} |U(x, t, \theta) - z_l(x, t, \theta, W, b)|^2 \\ W^*, b^* &= \underset{W, b}{\operatorname{argmin}} L(W, b) \end{aligned} \quad (3)$$

Here, $L(W, b)$ represents the loss of the training data, W^*, b^* denotes the weights and biases that need to be optimized during the neural network training process, and N indicates the number of training data samples.

Although FCNNs have strong nonlinear representation capabilities and can learn complex nonlinear problems, increasing the model's depth often leads to issues such as vanishing or exploding gradients. These problems cause network degradation during the later stages of training, making it difficult for the network to converge to an optimal solution. To address these challenges, He et al. [18] proposed the ResNet model, which achieved a breakthrough in training a very deep neural network, with hundreds of layers, by embedding shortcut connections into the FCNN structure. These shortcut connections allow the model to bypass one or more layers, enabling gradients to flow more easily through the network and mitigating the vanishing or exploding gradient problem. As a result, ResNet models can maintain training stability and improve performance even at significant depths.

The core of the ResNet model is its residual structure, as illustrated in Figure 1, which shows the output after passing through two hidden layers. In this structure, the input is defined as x , and the output as $H(x)$. The shortcut connection a key component of the residual structure adds the input of the network directly to the output obtained after passing through the hidden layers. This output is then processed by an activation function. This design significantly alleviates the issues of vanishing and exploding gradients during training, providing a more robust foundation for the model's learning accuracy and convergence. By allowing the gradients to flow more easily through the network, the shortcut connections enable deeper networks to be trained more effectively. The introduction of

ResNet has opened new directions for the development of deep neural networks, providing a powerful solution to the challenges encountered when training very deep networks.

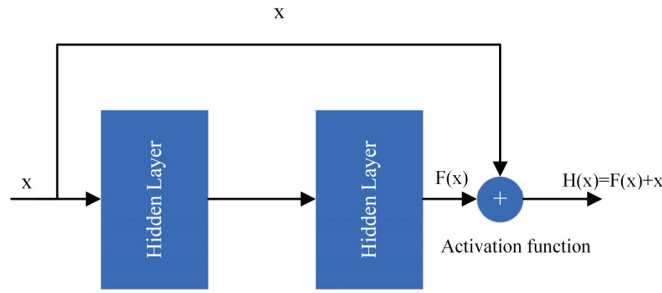


Figure 1. Structure of the ResNet neural network.

2.2. PINN Model

The PINN model is a type of neural network that incorporates physical knowledge into its framework, as shown as Figure 2. It enhances problem-solving capabilities by adding a physics-based loss function term to the data loss function of an FCNN. This additional loss term is derived from physical equations, ensuring that the predictions made by the neural network conform to the laws of physics during training. In the study of transient fluid flow characteristics, the PINN can reconstruct changes in the flow field state by updating the neural network with additional fluid control equations. This approach enables the network to account for the underlying physics governing fluid dynamics, leading to more accurate and physically consistent predictions. Its loss function can be defined as follows:

$$\begin{aligned}
 L_{PINN}(W, b) &= \lambda_{DATA} L_{DATA} + \lambda_{IC} L_{IC} + \lambda_{BC} L_{BC} + \lambda_{EQ} L_{EQ} \\
 L_{IC} &= \frac{1}{N_{IC}} \sum_{n=1}^{N_{IC}} \|U_n - \dot{U}_n\| \\
 L_{BC} &= \frac{1}{N_{BC}} \sum_{n=1}^{N_{BC}} \|U_n - \dot{U}_n\| \\
 L_{DATA} &= \frac{1}{N_{DATA}} \sum_{n=1}^{N_{DATA}} \|U_n - \dot{U}_n\| \\
 L_{EQ} &= \frac{1}{N_r} \sum_{n=1}^{N_r} \|r_n\|
 \end{aligned} \tag{4}$$

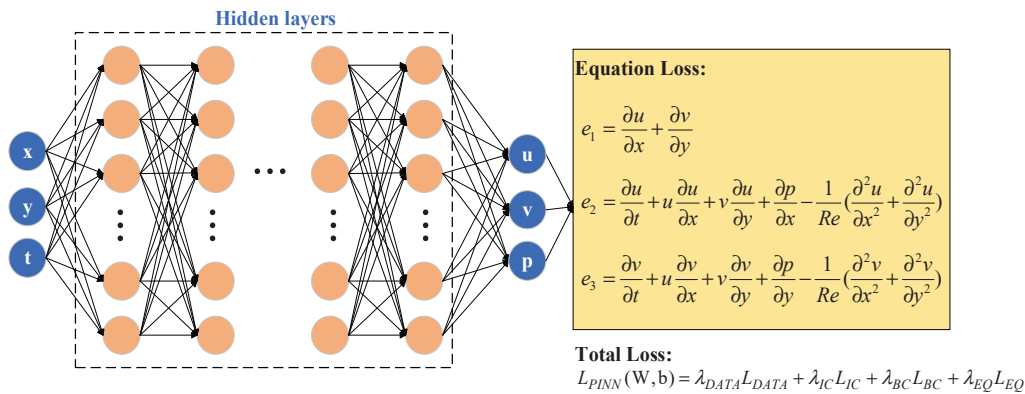


Figure 2. Structure of the PINN neural network model.

Here, λ_{DATA} , λ_{IC} , λ_{BC} , λ_{EQ} represents the coefficients for the loss functions corresponding to the different components, such as the training data, initial conditions, boundary

conditions, and the physical knowledge equation. These coefficients determine the relative importance of each component in the total loss function. N_R , N_{IC} , N_{BC} , N_{DATA} denotes the training data for the different terms, which may include observed values or measurements that the neural network aims to learn from U_N , $\dot{U}_N$ represents the exact or true results, as well as the results predicted by the neural network model. The difference between these two is minimized during training to improve the model's accuracy. r_n indicates the residual of the r -th physical knowledge equation at the n -th data point.

While the PINN offers a promising approach for constructing an ROM of fluid systems, it often encounters network “degradation” during training [19]. This degradation occurs because the physical constraints involved typically exhibit high diversity and complex nonlinear characteristics, which can cause sharp fluctuations in the gradients of the loss function in certain regions, resulting in extreme maxima or minima. Such fluctuations can destabilize the gradients, leading to issues like vanishing or exploding gradients during the training process. Moreover, when dealing with high-dimensional physical problems, the PINN must learn complex nonlinear mappings, further increasing the difficulty of training. The nature of these high-dimensional problems makes the neural network prone to “flat regions” or “saddle points”, where the gradient values are very small or nearly zero, preventing the model from effectively learning the critical features of the input data. Conversely, in regions where the gradients suddenly increase, the model may over-adjust its parameters, leading to inaccurate learning outcomes. These phenomena not only reduce the model's accuracy and stability but also significantly increase training time and computational costs. Therefore, to enhance the performance of PINNs, effective strategies must be employed to mitigate these challenges.

2.3. Res-PINNs Model

To effectively address the problem of performance degradation during neural network training and to achieve comprehensive and accurate extraction of flow field state information, this study proposes a multidimensional parallel PINN neural network, termed Res-PINNs. This model is based on the integration of the PINN and the ResNet model. The overall framework of the network structure is illustrated in Figure 3.

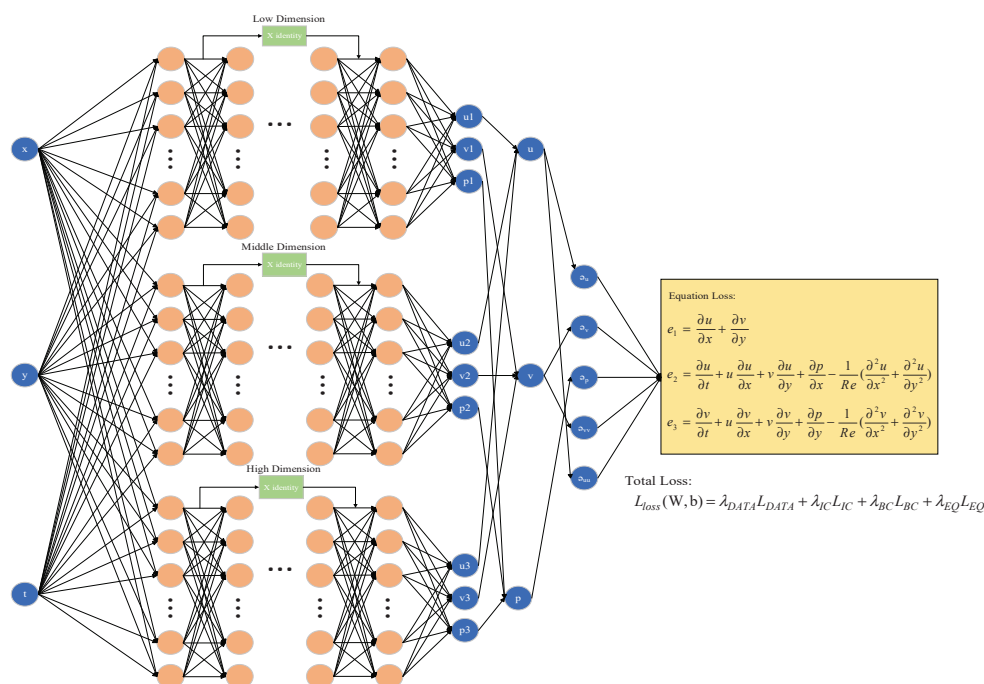


Figure 3. Structure of the Res-PINNs neural network.

According to the findings of Jagtap [20], segmenting the training data can reduce the overall loss function value, allowing for better extraction of flow field features and more accurate predictions of system changes. Inspired by previous studies, this study proposes an innovative architecture based on a symmetric parallel neural network to address the challenges posed by the complexity of flow field data. The network structure consists of three symmetric parallel neural networks, each responsible for processing flow field data at different dimensional levels: low, middle, and high. This symmetric network design effectively avoids the computational cost and time overhead associated with training large-scale, high-dimensional data simultaneously, while significantly enhancing the stability and robustness of the model during training. During the training process for each dimension, the individual neural networks operate independently. Thus, if one network encounters issues such as gradient vanishing or explosion, the other networks can continue training normally, preventing problems in one dimension from negatively affecting the entire model. To further optimize data processing, this study employs a Random Split (RS) algorithm to partition the spatial grid coordinates. The RS algorithm introduces randomness in partitioning by generating random indices based on a specified seed, which effectively reduces potential patterns or sequences in the data that could interfere with model training. In this study, the inflow velocity of the velocity field serves as a reference for partitioning, dividing the spatial grid coordinates into low-dimensional, middle-dimensional, and high-dimensional groups. During training, flow field state information is fed into the corresponding neural network models, which process the data according to their designated dimensional levels. To enhance the model's expressiveness and accuracy, this study incorporates a ResNet structure into the FCNN. This addition enables the network to more effectively capture the complex features of the flow field across different dimensions. The outputs from the networks at each dimensional level are integrated to represent the overall flow field state. This multidimensional training and output integration strategy allows the model to more comprehensively and accurately capture the intricate features of the flow field, thus significantly improving its performance. The proposed approach effectively addresses the challenges of processing high-dimensional flow field data, demonstrating superior accuracy and stability compared to traditional methods. The limited-memory Broyden Fletcher Goldfarb Shanno (L-BFGS) optimization algorithm, a quasi-Newton method, is designed for solving large-scale unconstrained optimization problems. Unlike gradient descent algorithms, L-BFGS does not require a global learning rate; it automatically adjusts the step size at each iteration using second-order information, making the optimization process more robust and automated. In this study, the loss function, established using both training data and physical equations, is employed to compute the loss [21]. The Adam optimizer and L-BFGS algorithm are used to perform backpropagation and parameter optimization, completing a single training cycle of the neural network. Notably, the loss values at the flow field boundary points are only trained in the low-dimensional neural network, and the initial and training data input into the three neural networks do not overlap. This neural network framework not only enables better extraction and learning of flow field feature information but also improves the model's performance during loss computation, reducing occurrences of gradient vanishing and explosion during training. It offers a useful approach and perspective for developing reduced-order models and applications in fluid systems.

3. Application of Res-PINNs

This section is divided by subheadings. It provides a concise and precise description of the experimental results, their interpretation, and the experimental conclusions that can be drawn.

3.1. Application of Res-PINNs to the Flow Around a Cylinder Problem

The flow around a cylinder problem is a classic case in Computational Fluid Dynamics (CFD), where different physical phenomena emerge in the flow field at various Reynolds numbers. In this case study, the objective is to reconstruct the flow field state of a two-dimensional flow around a cylinder using limited, discrete flow field data and a neural network model embedded with physical knowledge. A ROM for the flow around the cylinder is established, and the reconstruction results of the proposed Res-PINNs model are compared with those of the standard PINN model to validate the effectiveness of the network structure. In the flow around a cylinder problem, the Navier–Stokes (N-S) equations and the continuity equation typically serve as constraints. For two-dimensional, unsteady, incompressible fluid flow, the governing equations are generally expressed as follows:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (5)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0 \quad (6)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) = 0 \quad (7)$$

Here, (x, y) are the spatial coordinates of points in the flow field, and t represents time. The variables u and v denote the velocity components of the fluid: $u(x, y, t)$ is the velocity in the direction of the fluid inflow (denoted as IL), and $v(x, y, t)$ is the velocity perpendicular to the fluid inflow direction (denoted as CF). The variable p represents the absolute pressure at each spatial point in the flow field, $p(x, y, t)$. The term Re stands for the Reynolds number, which is a dimensionless parameter characterizing the flow regime, indicating the relative influence of inertial forces to viscous forces in the fluid.

In this case study, the numerical simulation method from CFD is used to obtain training data samples. The computational domain is illustrated in Figure 4, where the overall flow field region is a rectangular area. The left side serves as the velocity inlet, while the right side is the pressure outlet. The upper and lower boundaries are set with no-slip boundary conditions. The inlet velocity on the left is set to $u_0 = 0.09$ m/s, and the absolute pressure at the right outlet is 0 Pa. The dynamic viscosity of the fluid is $\mu = 1.5 \times 10^{-5}$ kg/m·s, the density is set to 1 kg/m³, and the cylinder diameter is $D = 0.05$, resulting in a Reynolds number of $Re = 300$; we chose the laminar flow model in the Ansys simulation model for the computational analysis.

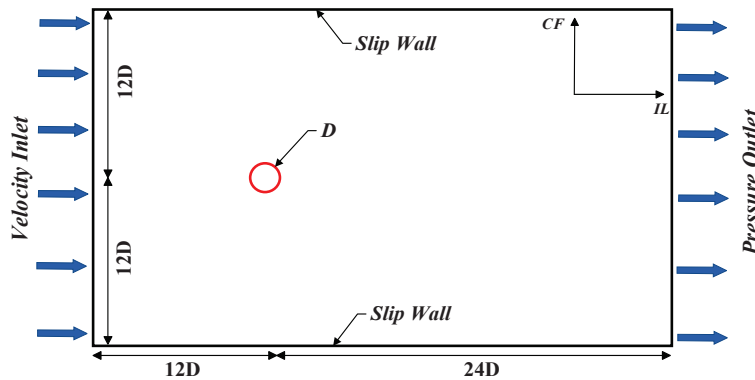


Figure 4. Computational domain for the flow around a cylinder problem.

An O-type grid method was employed for the grid partitioning of the flow field region, as show in Figure 5. The grid cell size was specified as 7.5 mm, and the height of the first

layer in the boundary layer grid was set to 2 mm, with a total of seven layers and a growth rate of 1.15. The computational domain was divided into 40,064 cells in total. According to the grid quality evaluation results, 98.08% of the grids were classified as high quality. This grid partitioning approach ensures both the accuracy and stability of the calculations, providing a solid foundation for subsequent numerical simulations. The simulation time step is set to 0.005 s, and the PISO algorithm is used for transient solution calculations.

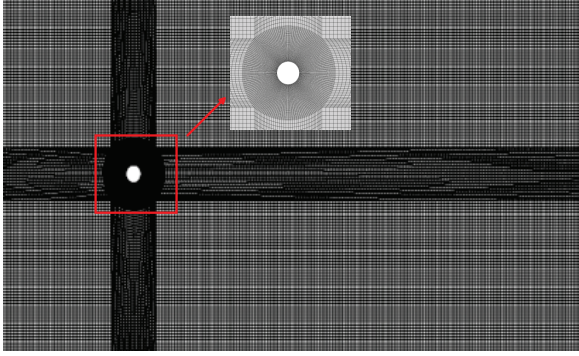


Figure 5. Mesh generation results for the flow field.

After 50 s from the start of the simulation, a stable vortex shedding phenomenon is observed. The flow field state information between 60 to 65 s is selected as the data collection period for training purposes. A training dataset is constructed using time slices of 0.1 s each. The spatial coordinates at each time instant are discretized, and the Latin hypercube sampling (LHS) algorithm is used to sample the spatial coordinates of the flow field region. During random sampling of the data, each time slice is processed independently. For each time slice, 1000 coordinate points are randomly sampled, and the corresponding flow field state information is used as training data for input into the training model. The data are then reset to a time frame of 0 to 5 s. The sampling results are shown in Figure 6.

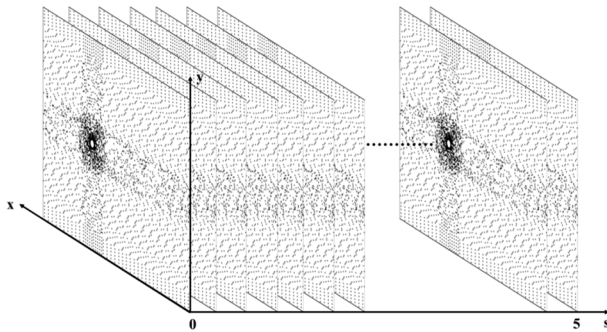


Figure 6. Results of LHS.

For both the PINN model and the Res-PINNs model, the input consists of the spatiotemporal coordinates $\{t_n, x_n, y_n\}$ of the sparse flow field, and the output includes the velocity and pressure information $\{p_n, u_n, v_n\}$ at each coordinate point within the flow field. After transformation, the equation for the loss function is expressed as follows:

$$\begin{aligned} e_1 &= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \\ e_2 &= \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ e_3 &= \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \end{aligned} \quad (8)$$

The total loss function is composed of two parts: the loss for the flow field state information and the loss for the governing control equations. The flow field state information includes the velocity components in the *IL* direction and *CF* direction, as well as the pressure information. The total loss function can be expressed as follows:

$$L_{data} = \lambda_{IN} \frac{1}{N_{IN}} \sum_{n=1}^{N_{IN}} |u(x_n, y_n, t_n) - u^n|^2 + \lambda_{IC} \frac{1}{N_{IC}} \sum_{n=1}^{N_{IC}} |u(x_n, y_n, t_n) - u^n|^2 + \lambda_{BC} \frac{1}{N_{BC}} \sum_{n=1}^{N_{BC}} |u(x_n, y_n, t_n) - u^n|^2 \quad (9)$$

$$L_{equ} = \lambda_{equ} \sum_{i=1}^3 \sum_{n=1}^N |e_i|^2$$

In the above expressions, N_{IN} , N_{IC} , N_{BC} denotes the grid points within the flow field, at the initial time, and along the boundary. $u(x_n, y_n, t_n)$, u^n represents the data obtained from neural network training and the reference data. L_{equ} denotes the residuals of the governing equations. λ_{IN} , λ_{IC} , λ_{BC} , λ_{equ} represents the flow field information at interior points, initial time flow field information, boundary point flow field information, and the weights for the loss function. Based on the analysis results from reference [22], the loss function weights for the Res-PINNs neural network are set to $\lambda_{IN} = 1$, $\lambda_{IC} = 1$, $\lambda_{BC} = 1$, $\lambda_{equ} = 75$, while the loss function weights for the PINN neural network are set to $\lambda_{IN} = 1$, $\lambda_{IC} = 1$, $\lambda_{BC} = 1$, $\lambda_{equ} = 150$. These loss functions are then incorporated into the neural networks for model training. To avoid overfitting during the network training process, a dropout layer was added before the network output to reduce the interdependence between neurons, with the dropout rate set to 0.3. The DNN are trained using the machine learning framework TensorFlow on an NVIDIA GeForce RTX 3080 Ti GPU. After approximately 11 h of training, the model converged. Figure 7 shows the training loss as a function of the number of training steps. After 14,000 iterations, the training error converged to less than 0.01.

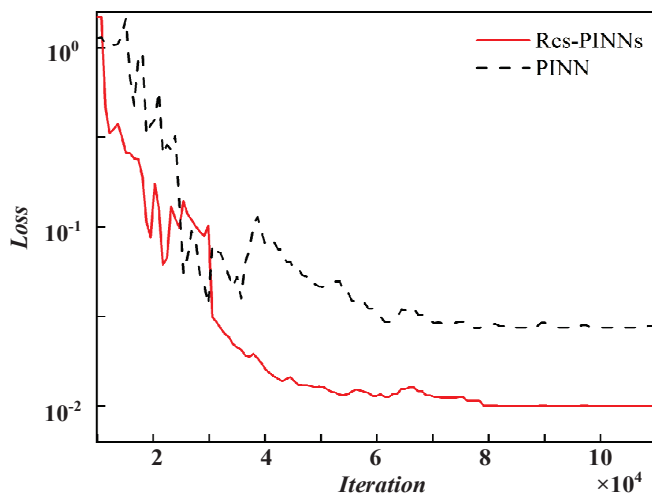


Figure 7. Results of the model loss function.

In this case study, the model's effectiveness is evaluated by reconstructing the flow field state at 5.5 s using sparse flow field information learned from 0 to 5 s. The reconstruction results are shown in Figure 8, where Figure 8a presents the reconstruction results using the PINN model; Figure 8b shows the reconstruction results using the Res-PINNs model. Within both Figure 8a,b, the first column displays the simulation results obtained from Ansys Fluent, which serve as a reference for evaluating the model's accuracy. The second column shows the reconstructed flow field results from the ROM. The third column illustrates the error between the reference simulation results and the ROM reconstructions.

From top to bottom, each row represents the pressure field contour plot, the velocity field contour plot in the IL direction, and the velocity field contour plot in the CF direction.

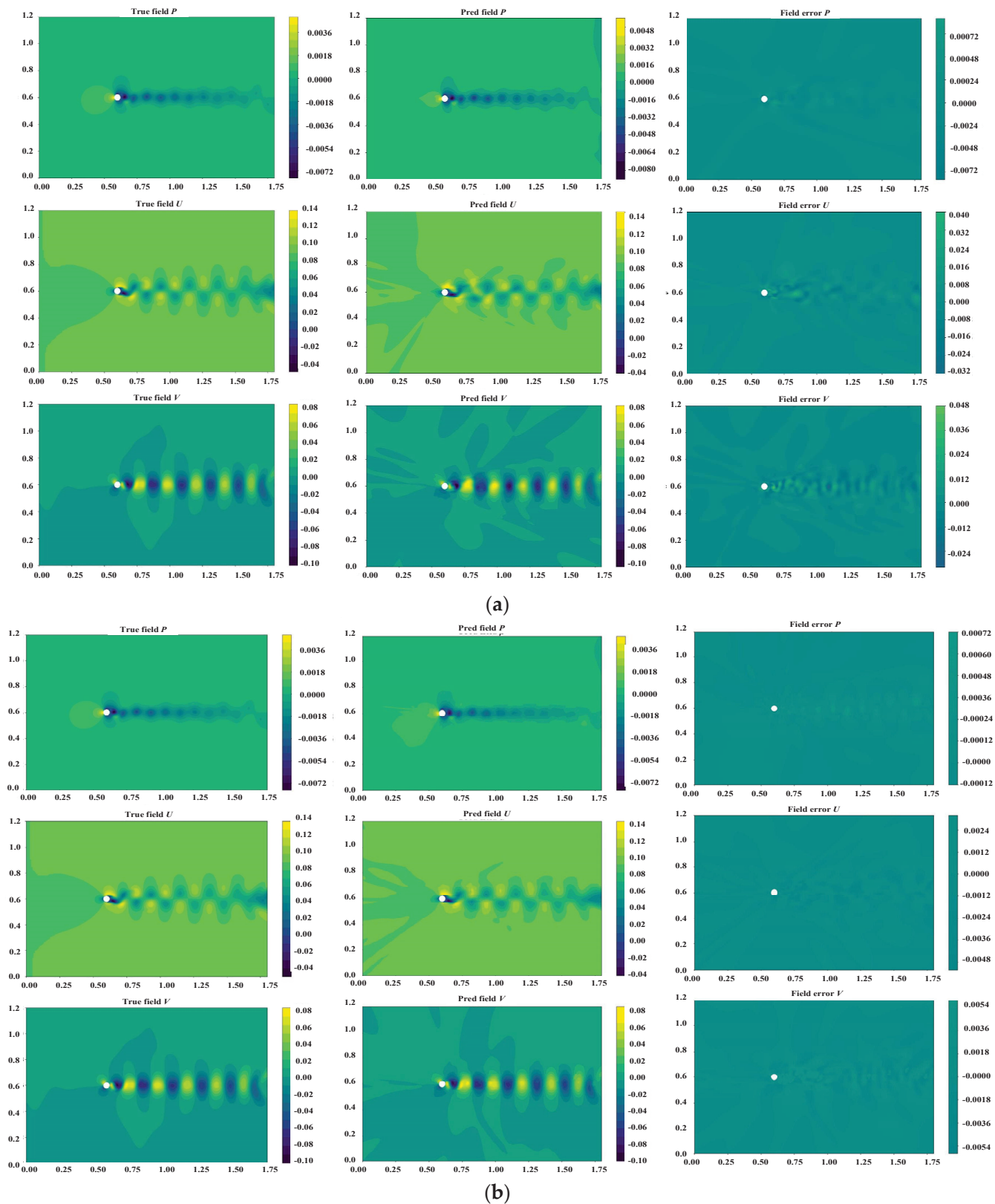


Figure 8. Flow field reconstruction results of PINN and Res-PINNs model. (a) PINN reconstruction results; (b) Res-PINNs reconstruction results.

Based on the computational analysis of the results, for the PINN model, the relative error in the pressure field reconstruction is 10.9%; the relative errors in the velocity fields along the *IL* and *CF* directions are 38.8% and 39.2%, respectively; and the larger errors in the velocity fields are mainly concentrated around the cylinder wall region and the wake region. This discrepancy may be due to gradient vanishing or exploding during training, causing minimal changes in the loss function value as the number of iterations increases. Consequently, the network becomes less sensitive to learning the flow field states. In contrast, for the Res-PINNs model proposed in this study: The relative error in the pressure field is reduced to 7.8%; the relative errors in the velocity fields along the *IL* and *CF* directions are significantly reduced to 4.1% and 6.3%, respectively; and the Res-PINNs model demonstrates a more accurate reconstruction of the flow field state, particularly around the cylinder and in capturing vortex shedding features. This indicates that the model achieves more comprehensive learning of the flow characteristics, offering more reliable results for flow field modeling and reconstruction.

Overall, the results show that the Res-PINNs model significantly outperforms the standard PINN model in terms of accuracy and error reduction, particularly in regions with complex flow features. This highlights the effectiveness of the Res-PINNs framework in providing more accurate and stable solutions for fluid dynamics problems.

The analysis above indicates that the Res-PINNs model performs significantly better than the standard PINN model when dealing with complex flow fields. The Res-PINNs model demonstrates a more sensitive and comprehensive ability to capture state changes within the flow field. Although there are still some errors in certain regions of the flow field, overall, the Res-PINNs model exhibits a superior performance in flow field reconstruction compared to the standard PINN model.

Based on the reconstructed flow field data, the pressure and velocity gradient distributions on the cylinder surface can be integrated to compute the lift coefficient and drag coefficient of the cylinder. These coefficients are key parameters in characterizing the aerodynamic forces acting on the cylinder. The formulas for calculating these coefficients are as follows:

$$\begin{aligned} F_L &= \int_{\Gamma} \left[\frac{1}{Re} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) n_x + \frac{2}{Re} \frac{\partial v}{\partial y} n_y - p n_y \right] ds \\ F_D &= \int_{\Gamma} \left[\frac{1}{Re} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) n_y + \frac{2}{Re} \frac{\partial u}{\partial x} n_x - p n_x \right] ds \end{aligned} \quad (10)$$

In the equations, Γ represents the boundary of the cylinder's surface. n_x and n_y are the components of the unit normal vector on the cylinder surface. The dimensionless formulas for calculating the lift coefficient and drag coefficient are as follows:

$$C_{L/D} = \frac{F_{L/D}}{0.5 \rho u_0^2 D} \quad (11)$$

Figure 9 illustrates the reconstruction results of the pressure and lift coefficient based on the flow field data. In the figure, the black line represents the results obtained from CFD simulation. The blue line indicates the predicted values from the Res-PINNs model. The red line shows the predicted results from the PINN model. From the results, it can be observed that for the prediction of C_D , the ROM results have a smaller error compared to the CFD simulation results. In contrast, the PINN model shows a larger amplitude difference compared to the CFD simulation results. This may be due to the gradient vanishing problem in the neural network model during the flow field reconstruction process, making it less sensitive to changes in the flow field state around the cylinder wall and failing to learn relevant flow characteristics, resulting in biased predictions in subsequent processes.

The Res-PINNs prediction results are closer to the CFD simulation results; for the prediction of C_L , both the PINN and Res-PINNs models show some phase differences compared to the CFD simulation results. However, the Res-PINNs model, being more sensitive to changes in the flow field gradients, yields predictions that are closer to the CFD simulation results compared to the PINN model. The above analysis leads to the conclusion that, compared to the PINN model, the results of the flow field state reconstructed by the Res-PINNs model are in better agreement with the CFD simulation results.

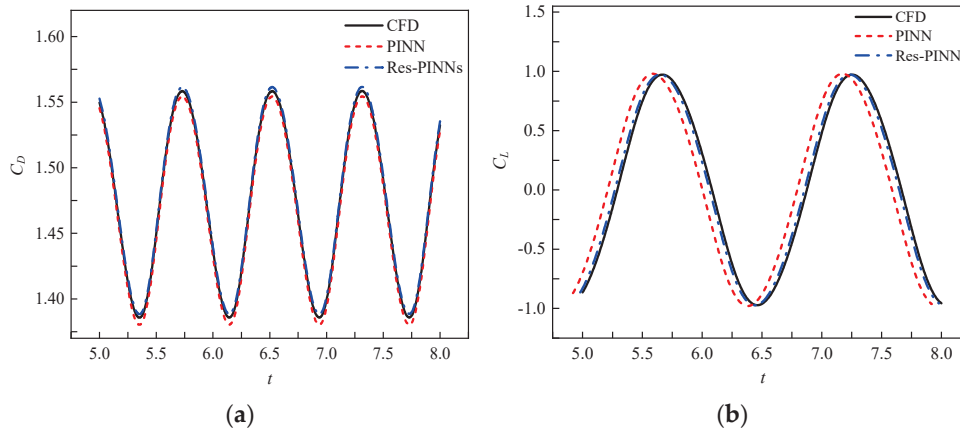


Figure 9. Reconstruction results of flow field lift and drag: (a) C_D reconstruction results; (b) C_L reconstruction results.

3.2. Application of Res-PINNs to the Vortex-Induced Vibration Problem

VIV is a coupled fluid–structure interaction problem that occurs when a structure is placed in a flow field at a certain velocity. Periodic pulsating fluid forces along the flow direction and perpendicular to it cause the elastic structure to undergo periodic oscillations. This interaction alters the vortex shedding pattern of the fluid, resulting in coupled variations in fluid flow, structural displacement, and forces acting on the structure.

In this case study, the Reynolds averaged Navier–Stokes (RANS) equations are embedded into the Res-PINNs framework to solve high Reynolds number VIV problems, validating the model’s effectiveness. An incompressible fluid is governed by the N-S equations and the mass conservation equation, as detailed in Section 3.1. However, as the Reynolds number increases, turbulence becomes significant, which not only raises the computational cost but can also lead to ill-conditioned matrices during numerical solutions. To address the challenges posed by turbulence, the RANS equations are introduced. The RANS equations provide a time-averaged approach to deal with turbulent flow, reducing the complexity of modeling turbulence while retaining the essential flow characteristics. The equations are expressed as follows:

$$\begin{aligned}\frac{\partial \bar{u}}{\partial t} + \bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} &= -\frac{\partial \bar{p}}{\partial x} + (\nu + \nu_t) \left(\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} \right) \\ \frac{\partial \bar{v}}{\partial t} + \bar{u} \frac{\partial \bar{v}}{\partial x} + \bar{v} \frac{\partial \bar{v}}{\partial y} &= -\frac{\partial \bar{p}}{\partial y} + (\nu + \nu_t) \left(\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} \right) \\ \frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} &= 0\end{aligned}\quad (12)$$

Here, ν_t represents the turbulent eddy viscosity coefficient. The RANS equations simulate turbulence by establishing a relationship between the eddy viscosity coefficient and the time-averaged parameters of the turbulence. The structural vibration equation can

be modeled using a typical mass-spring-damper system. The equations of motion for the cylinder in the *IL* direction and the *CF* direction can be expressed as follows:

$$\begin{aligned} m\ddot{\zeta} + c\dot{\zeta} + k\zeta &= F_D(t) \\ m\ddot{\eta} + c\dot{\eta} + k\eta &= F_L(t) \end{aligned} \quad (13)$$

In these equations, m represents the mass of the cylinder. c and k denote the damping coefficient and stiffness coefficient of the cylinder, respectively. $F_L(t)$ and $F_D(t)$ represent the lift and drag forces. ζ , η indicate the displacements along the *IL* and *CF* directions. At the initial state, the boundary conditions for the cylinder are as follows:

$$\begin{aligned} \dot{\zeta}(0) &= \dot{\eta}(0) = 0 \\ \zeta(0) &= \eta(0) = 0 \end{aligned} \quad (14)$$

By discretizing the structural equations using the fourth-order Runge–Kutta method, the velocity and displacement responses of the cylinder can be solved. For solving the velocity and displacement in the *IL* direction, the equations can be expressed as follows:

$$\begin{aligned} \dot{\zeta}(t_{n+1}) &= \dot{\zeta}(t_n) + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ \zeta(t_{n+1}) &= \zeta(t_n) + \dot{\zeta}(t_n)\Delta t + \frac{\Delta t^2}{6}(k_1 + k_2 + k_3) \end{aligned} \quad (15)$$

Among them,

$$\begin{aligned} k_1 &= \frac{F_D(t_n)}{m} - \frac{c}{m}\dot{\zeta}(t_n)\Delta t - \frac{k}{m}\zeta(t_n) \\ k_2 &= \frac{F_D(t_n)}{m} - \frac{c}{m}\left[\dot{\zeta}(t_n) + \frac{\Delta t}{2}k_1\right] - \frac{k}{m}\left[\zeta(t_n) + \frac{\Delta t}{2}\dot{\zeta}(t_n)\right] \\ k_3 &= \frac{F_D(t_n)}{m} - \frac{c}{m}\left[\dot{\zeta}(t_n) + \frac{\Delta t}{2}k_2\right] - \frac{k}{m}\left[\zeta(t_n) + \frac{\Delta t}{2}\dot{\zeta}(t_n) + \frac{\Delta t^2}{4}k_1\right] \\ k_4 &= \frac{F_D(t_n)}{m} - \frac{c}{m}\left[\dot{\zeta}(t_n) + \frac{\Delta t}{2}k_3\right] - \frac{k}{m}\left[\zeta(t_n) + \frac{\Delta t}{2}\dot{\zeta}(t_n) + \frac{\Delta t^2}{2}k_2\right] \end{aligned} \quad (16)$$

In the formula, k_1 , k_2 , k_3 , and k_4 are the coefficients used in the Runge–Kutta method, and Δt represents the simulation time step. The displacement and velocity responses along the *CF* direction are solved in a manner similar to the equations for the *IL* direction.

In the VIV case, CFD simulation technology is used to obtain the training data. The overall flow field is modeled and simulated using the Ansys Fluent software, and a schematic diagram of the entire flow field region is shown in Figure 10. The structure consists of a 2-DOF two-dimensional cylindrical body with a spring. The left side is a velocity inlet with a magnitude of $u_0 = 1$ m/s, the right side is a 0 MPa absolute pressure outlet, the upper and lower boundaries are no-slip boundaries, and the cylinder surface is a no-slip wall. The overset mesh region is set to a size of $3D$, and an unstructured *O-type* mesh is used to divide the refined region. When the mesh is updated, the internal region of the *O-type* mesh remains unchanged during the movement of the cylinder, while the area outside the *O-type* mesh changes shape. The mesh division results are shown in Figure 11. The size of the entire computational domain is $18D \times 12D$. The grid cell size was set to 3 mm, with the height of the first layer set to 1 mm and a growth rate of 1.3. The grid consisted of 60,040 cells in total, with 97.32% of the grids being categorized as high quality, according to the grid quality evaluation results. The Reynolds number is set to $Re = 1000$, with a fluid density of 1 kg/m^3 and a dynamic viscosity of $\mu = 0.5 \times 10^{-4} \text{ kg/(m}\cdot\text{s)}$. The cylinder's mass is set to $m = 2 \text{ kg}$, the damping coefficient $c = 0.084$, and the stiffness $k = 2.2020$. During the simulation, the velocity, pressure, and other information at each grid point in the flow field

are first computed, and the forces acting on the cylinder surface are calculated. This force information is then transferred to a compiled UDF, where the fourth-order Runge–Kutta method is used to solve the equations of motion for the structure. This process determines the displacement and velocity of the cylinder at the next time step, and the results are fed back into Fluent to update the flow field state for the subsequent time-step calculations, and the simulation time step is set to 0.005 s. The simulation model uses the *SST $k-\omega$* model, primarily because this model provides relatively accurate turbulence predictions in complex flow problems, especially in the near-wall region and in capturing flow separation. Compared to DNS and experimental flow types, the *SST $k-\omega$* model offers a reasonable trade-off between computational efficiency and accuracy.

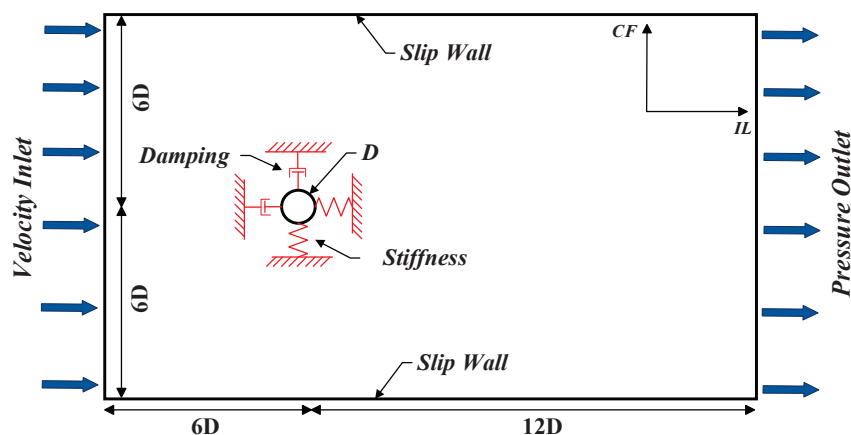


Figure 10. Computational domain for the VIV problem.

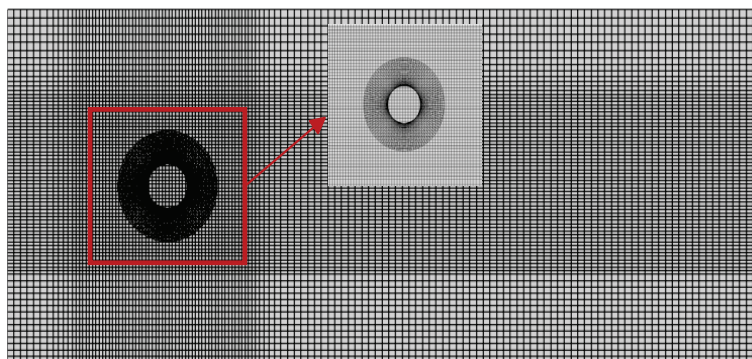


Figure 11. Mesh division results of the VIV flow field.

Once the VIV motion of the cylinder reaches a steady state, the velocity field, pressure field, and cylinder motion information are extracted from the CFD simulation results as training data. The period from 20 to 25 s, after the motion has stabilized, is selected as the training dataset, with a time sampling interval of 0.1 s. To ensure that the data used are sufficient to support model training, we focused on its diversity and coverage during the data collection process, a Latin hypercube sampling method is employed to sample the flow field region, and 7000 data points are chosen as training data for each time step [1].

In this case, the ROM based on Res-PINNs is used to reconstruct the entire flow field state using sparse flow field information. The neural network structure adopts the same architecture as the previous case, with the network input being the spatiotemporal data $\{t_n, x_n, y_n\}$ of the sparse flow field. When solving for VIV, due to the coupling between the structure and the fluid, the structural vibration is also considered as a feature for both training and prediction. Additionally, the turbulent eddy viscosity coefficient is treated as

an unknown parameter to be solved, making the network output $\{p_n, u_n, v_n, \varepsilon_n, \eta_n, v_t\}$. For simplicity, the loss function of this network omits the overline notation from Equation (12). By embedding the horizontal and vertical displacements of the cylinder into Equation (12), the equation's loss function is formulated as shown in Equation (17).

$$\begin{aligned} e_1 &= \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - (\nu + \nu_t) \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial^2 \zeta}{\partial t^2} \\ e_2 &= \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - (\nu + \nu_t) \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{\partial^2 \eta}{\partial t^2} \\ e_3 &= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \end{aligned} \quad (17)$$

The overall loss function of the neural network model is the sum of the training data loss and the equation loss, as shown in Equation (18).

$$\begin{aligned} L_{data} &= \sum_{n=1}^N |u(x_n, y_n, t_n) - u_n|^2 + \sum_{n=1}^N |v(x_n, y_n, t_n) - v_n|^2 + \sum_{n=1}^N |p(x_n, y_n, t_n) - p_n|^2 + \sum_{n=1}^N |v_t(x_n, y_n, t_n) - v_t^n|^2 \\ &+ \sum_{n=1}^N |\eta(t_n) - \eta_t^n|^2 + \sum_{n=1}^N |\zeta(t_n) - \zeta_t^n|^2 \\ L_{equ} &= \sum_{i=1}^3 \sum_{n=1}^N |e_i|^2 \end{aligned} \quad (18)$$

where $u(x_n, y_n, t_n)$, $v(x_n, y_n, t_n)$, $p(x_n, y_n, t_n)$, $v_t(x_n, y_n, t_n)$, and $\zeta(t_n)$, $\eta(t_n)$ represent the output results after neural network training. The loss is calculated, and parameter updates are performed through backpropagation using the Adam optimizer and the L-BFGS optimization method. This completes one training iteration of the neural network. Similarly, to prevent overfitting, a dropout layer was also added before the network output, with the dropout rate set to 0.5. The training is conducted using the same simulation environment and configuration as in Case 3.1. After 150,000 iterations, the network converges, and the loss calculation results are shown in Figure 12.

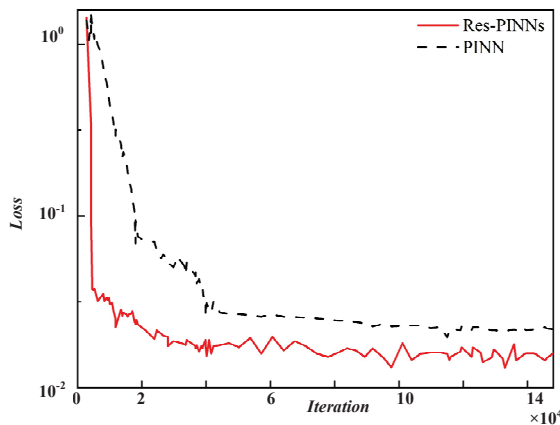


Figure 12. Model loss calculation and results.

The effectiveness of the model is evaluated by comparing the flow field reconstruction results at the 30 s mark using the ROM of Res-PINNs and PINN. The results are presented in Figure 13, where Figure 13a shows the reconstruction from the PINN model, and Figure 13b displays the reconstruction from the Res-PINNs model. In both figures, the first column provides the Ansys Fluent simulation results as a reference, the second column shows the reconstructed results from the ROM, and the third column illustrates the error between the reference simulation and the reconstructed results. From top to bottom, the rows represent the pressure field contour plot, the velocity field contour plot along the IL direction, and the

velocity field contour plot along the *CF* direction. Calculations of the prediction results for the two models reveal that the velocity field reconstructed by the Res-PINNs model shows errors of 17.7% and 17.2% relative to the CFD simulation results, while the errors for the PINN model are 40.3% and 30.8%, respectively. For the pressure field, the reconstruction errors of the Res-PINNs and PINN models are 14% and 35.1%, respectively. The areas with significant errors are primarily located in the *O-type* mesh and its vicinity. This discrepancy is likely due to the limitations of the PINN model's network architecture, which cannot fully analyze high-dimensional and nonlinear problems, resulting in a reduced ability to learn the flow field state features. In contrast, the Res-PINNs model, with its more complex network structure, can more accurately capture the velocity distribution of the flow field. The reconstruction results indicate that the Res-PINNs model outperforms the PINN model in the wake region, mitigating the shortcomings found in the PINN model.

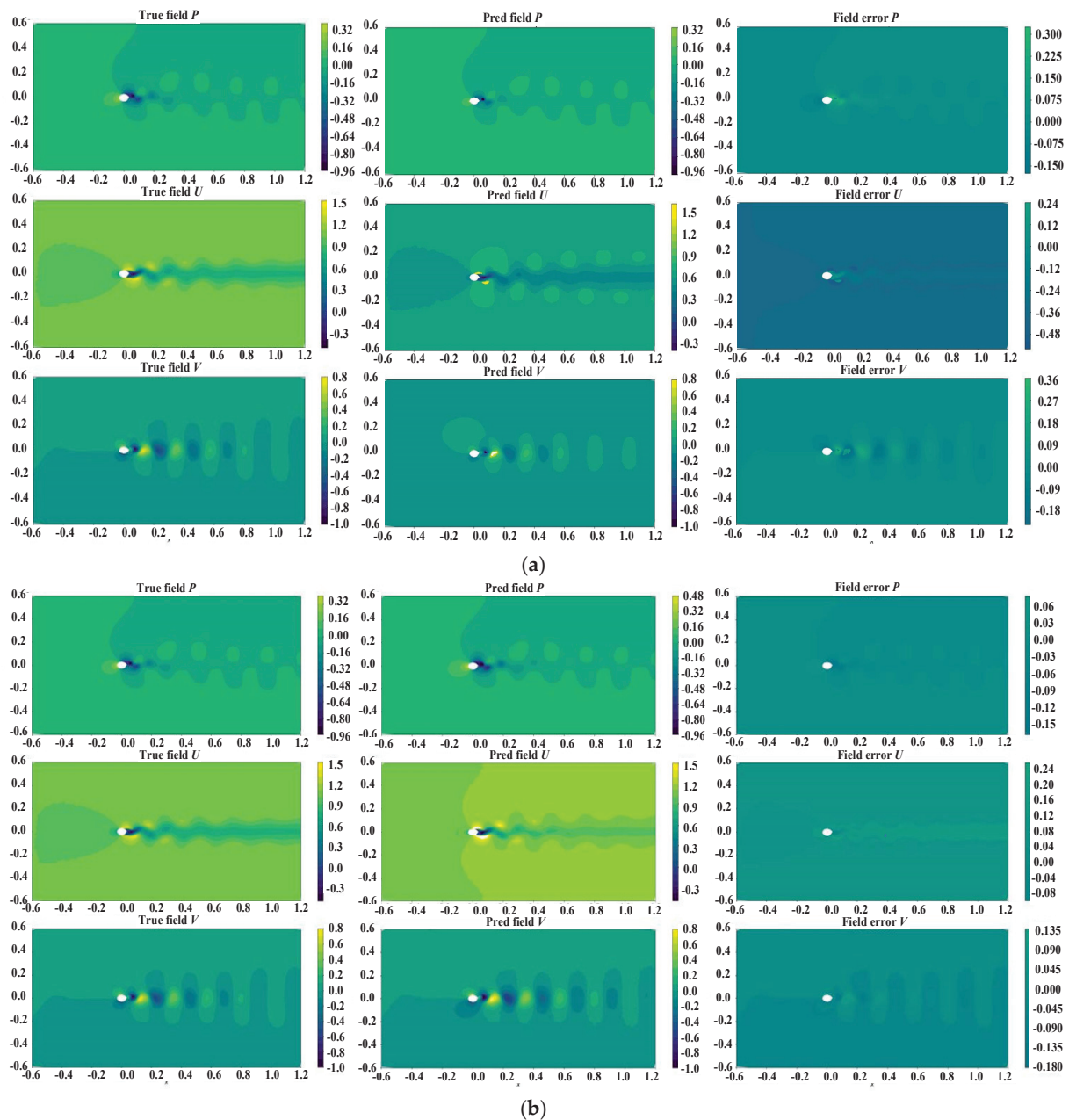


Figure 13. Flow field reconstruction results. (a) Reconstruction results using PINN; (b) reconstruction results using Res-PINNs.

Figure 14 shows the reconstruction results for the lift coefficient, drag coefficient, and structural displacement of the flow field from 25 to 28 s. In the figure, the black line represents the exact values obtained from the CFD simulation, the blue line represents the reconstruction results from the Res-PINNs model, and the red line represents the reconstruction results from the PINN model. Analysis of the reconstruction results reveals significant differences between the reconstructed structural vibration responses and the CFD simulation results. These discrepancies are more noticeable at the peaks and troughs of the response states, and there is a clear phase difference when compared to the CFD results. However, the overall trends remain consistent with the CFD results. A comparison with other studies [23,24] suggests that this discrepancy may result from the relatively low resolution of the data slicing, which impedes the model's ability to capture the dynamic coupling between the flow field and the structure. Consequently, the model fails to accurately capture the transient behavior of the flow field and structural dynamics, particularly during more intense dynamic responses. The insufficient temporal information further causes phase deviations in the prediction results along the time axis, impairing the model's ability to accurately track changes in the flow field. This discrepancy suggests that a greater amount of time-sliced data might be required for the model to learn and compute more accurately when constructing the training dataset. For the reconstruction of the lift and drag coefficients, the Res-PINNs model provides results that are more accurate and closer to the CFD simulation outcomes compared to the PINN model.

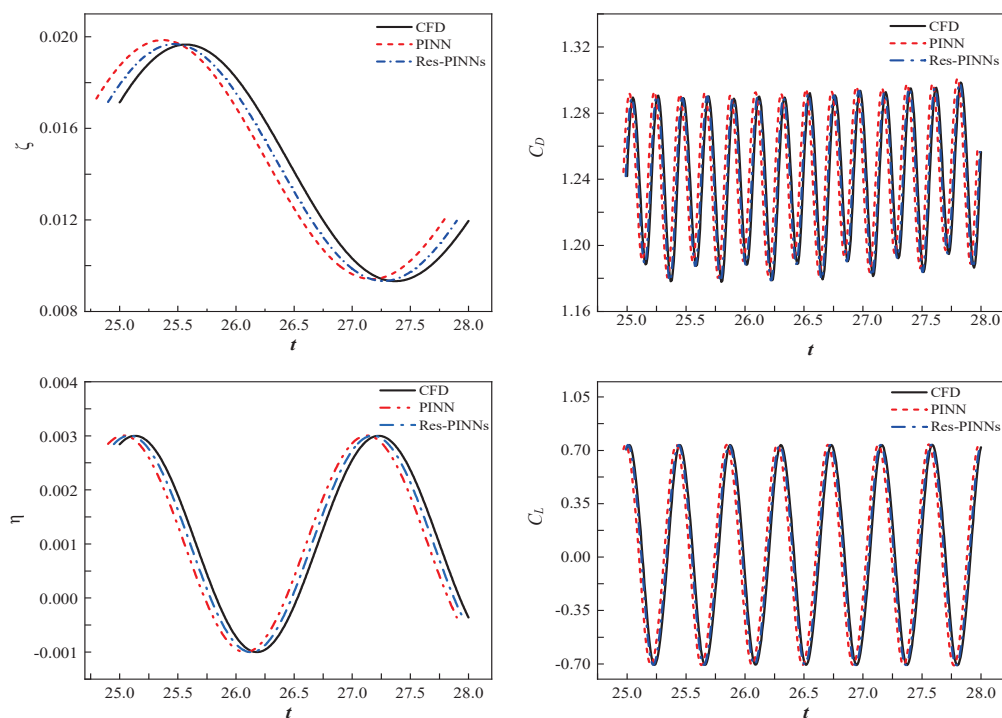


Figure 14. Displacement response and reconstruction results.

Based on the above analysis, it can be concluded that both the Res-PINNs and PINN models exhibit significant differences from the CFD simulation results when reconstructing the structural response in VIV problems. However, for the reconstruction of the lift and drag coefficients, the Res-PINNs model demonstrates superior performance.

4. Conclusions

This paper proposes a parallel Res-PINNs neural network, building on the PINN ROM, to achieve fast simulation and flow field reconstruction for fluid systems. Compared to

traditional PINN neural network models, the proposed Res-PINNs model integrates ResNet modules to address the issues of gradient vanishing or exploding that can occur during the training of FCNN. Additionally, the network structure has been optimized with a parallel network training strategy to better extract and learn flow field state information, thereby enhancing the quality of flow field learning and reconstruction. To validate the performance of the proposed Res-PINNs neural network, two cases were used for verification: a two-dimensional cylinder flow and a 2DOF VIV case at high Reynolds numbers. The results were compared with predictions from the traditional PINN neural network. The findings indicate the following:

- (1) For the two-dimensional cylinder flow problem, the Res-PINNs model predicted flow field velocity and pressure errors of 4.1%, 6.3%, and 7.8%, respectively, compared to 38.8%, 39.2%, and 10.9% for the PINN model. Additionally, the Res-PINNs model provided more accurate reconstruction results for the lift and drag coefficients, demonstrating its superior ability to handle complex flow fields and capture variations in flow field states more comprehensively and sensitively.
- (2) For the 2-DOF VIV case at high Reynolds numbers, the Res-PINNs model, which incorporates the RANS equations, the eddy viscosity coefficient, and structural motion equations, accurately predicted the VIV state information. The Res-PINNs model showed flow field velocity and pressure errors of 17.7%, 17.2%, and 14%, respectively, compared to 40.3%, 30.8%, and 35.1% for the PINN model. These results validate the effectiveness of the Res-PINNs model, although there remains room for improvement in reconstructing structural displacement responses accurately.
- (3) Moreover, for the same number of time steps, the Res-PINNs model requires only one-twentieth of the simulation time of numerical simulations, significantly enhancing computational efficiency.

5. Discussion

The proposed Res-PINNs model demonstrates significant potential in tackling challenges in fluid system modeling and simulation. However, the flow field reconstruction problem itself exhibits strong nonlinear characteristics, especially when the coupling effects between the fluid and structure are complex. Simplified ROMs may fail to capture all key dynamics, leading to an increase in error. Additionally, the quality and coverage of the training data are crucial factors influencing the model's accuracy. The sparsity of data slices and the insufficient number of data points may prevent the network from accurately capturing the transient behaviors of complex flow fields, particularly in cases with significant nonlinear dynamic responses. Thus, the prediction accuracy and generalization capability of the model still need further refinement. In future work, we will focus on optimizing the model structure to achieve more accurate flow field state distributions. While this study is limited to cylinder flow and VIV cases, future research should explore more complex problems to enhance the model's robustness and better address the practical demands of engineering applications.

Author Contributions: Conceptualization, Y.L.; Data curation, R.Z.; Formal analysis, Y.L.; Investigation, P.W.; Methodology, Y.L.; Resources, J.J.; Software, Y.L.; Supervision, R.Z.; Validation, J.H.; Visualization, R.Z.; Writing—original draft, Y.L.; Writing—review and editing, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by China State Administration of Science, Technology, and Industry for National Defense Civil Aerospace Project, grant number D020101.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data involves sensitive information and is related to confidential aspects of my research project, and therefore cannot be shared at this stage.

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

Abbreviations

PINNs	Physics-Informed Neural Networks
ROM	Reduced-Order Model
Res-PINNs	Physics-Informed Neural Networks based on Residual Networks
ResNet	Residual Network
VIV	Vortex Induced Vibration
M&S	Modeling and Simulation
POD	Proper Orthogonal Decomposition
DMD	Dynamic Mode Decomposition
AI	Artificial Intelligence
DL	Deep Learning
RNN	Recurrent Neural Network
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
PDEs	Partial Differential Equations
FCNN	Fully Connected Neural Network
ReLU	Rectified Linear Unit
RS	Random Split
L-BFGS	Limited-memory Broyden Fletcher Goldfarb Shanno
CFD	Computational Fluid Dynamics
N-S	Navier–Stokes
LHS	Latin Hypercube Sampling
RANS	Reynolds Averaged Navier–Stokes

References

1. Ding, H.; Shu, C.; Yeo, K.; Xu, D. Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method. *Comput. Methods Appl. Mech. Eng.* **2004**, *193*, 727–744. [CrossRef]
2. Liu, F.; Zheng, X. A Strongly Coupled Time-Marching Method for Solving the Navier–Stokes and $k-\omega$ Turbulence Model Equations with Multigrid. *J. Comput. Phys.* **1996**, *128*, 289–300. [CrossRef]
3. Lucia, D.J.; Beran, P.S.; Silva, W.A. Reduced-order modeling: New approaches for computational physics. *Prog. Aerosp. Sci.* **2004**, *40*, 51–117. [CrossRef]
4. Li, K.; Kou, J.; Zhang, W. Unsteady aerodynamic reduced-order modeling based on machine learning across multiple airfoils. *Aerosp. Sci. Technol.* **2021**, *119*, 107173. [CrossRef]
5. Fresca, S.; Dede', L.; Manzoni, A. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. *J. Sci. Comput.* **2021**, *87*, 61. [CrossRef]
6. Brahmachary, S.; Bhagyarajan, A.; Ogawa, H. Fast estimation of internal flowfields in scramjet intakes via reduced-order modeling and machine learning. *Phys. Fluids* **2021**, *33*, 106110. [CrossRef]
7. Jjana, A.; Mitra, A.S.; Das, S.; Chueh, W.C.; Bazant, M.Z.; García, R.E. Physics-based, reduced order degradation model of lithium-ion batteries. *J. Power Sources* **2022**, *545*, 231900. [CrossRef]
8. Janda, T.; Schmidt, J.; Hála, P.; Konrád, P.; Zemanová, A.; Sovják, R.; Zeman, J.; Šejnoha, M. Reduced order models of elastic glass plate under low velocity impact. *Comput. Struct.* **2021**, *244*, 106430. [CrossRef]
9. Halder, R.; Damodaran, M.; Khoo, B. Deep learning based reduced order model for airfoil-gust and aeroelastic interaction. *AIAA J.* **2020**, *58*, 4304–4321. [CrossRef]
10. Zhang, M.; Fu, S.; Ren, H.; Ma, L.; Xu, Y. A hybrid FEM-DNN-based vortex-induced Vibration Prediction Method for Flexible Pipes under oscillatory flow in the time domain. *Ocean Eng.* **2022**, *246*, 110488. [CrossRef]
11. Li, K.; Kou, J.; Zhang, W. Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers. *Nonlinear Dyn.* **2019**, *96*, 2157–2177. [CrossRef]

12. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
13. Raissi, M.; Wang, Z.; Triantafyllou, M.S.; Karniadakis, G.E. Deep learning of vortex-induced vibrations. *J. Fluid Mech.* **2019**, *861*, 119–137. [CrossRef]
14. Wang, J.-X.; Wu, J.; Ling, J.; Iaccarino, G.; Xiao, H. A comprehensive physics-informed machine learning framework for predictive turbulence modeling. *arXiv* **2017**, arXiv:1701.07102.
15. Bararnia, H.; Esmaeilpour, M. On the application of physics informed neural networks (PINN) to solve boundary layer thermal-fluid problems. *Int. Commun. Heat Mass Transf.* **2022**, *132*, 105890. [CrossRef]
16. Tang, H.; Liao, Y.; Yang, H.; Xie, L. A transfer learning-physics informed neural network (TL-PINN) for vortex-induced vibration. *Ocean Eng.* **2022**, *266*, 113101. [CrossRef]
17. Arzani, A.; Wang, J.-X.; D’Souza, R.M. Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Phys. Fluids* **2021**, *33*, 071905. [CrossRef]
18. 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.
19. Zhang, G.; Yang, H.; Pan, G.; Duan, Y.; Zhu, F.; Chen, Y. Constrained Self-Adaptive Physics-Informed Neural Networks with ResNet Block-Enhanced Network Architecture. *Mathematics* **2023**, *11*, 1109. [CrossRef]
20. Jagtap, A.D.; Karniadakis, G.E. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Commun. Comput. Phys.* **2020**, *28*, 2002–2041. [CrossRef]
21. Pratama, D.A.; Abo-Alsabeh, R.R.; Bakar, M.A.; Salhi, A.; Ibrahim, N.F. Solving partial differential equations with hybridized physic-informed neural network and optimization approach: Incorporating genetic algorithms and L-BFGS for improved accuracy. *Alex. Eng. J.* **2023**, *77*, 205–226. [CrossRef]
22. Kag, V.; Seshasayanan, K.; Gopinath, V. Physics-informed data based neural networks for two-dimensional turbulence. *Phys. Fluids* **2022**, *34*, 055130. [CrossRef]
23. Hijazi, S.; Freitag, M.; Landwehr, N.J.A.M.; Sciences, S. POD-Galerkin reduced order models and physics-informed neural networks for solving inverse problems for the Navier–Stokes equations. *Adv. Model. Simul. Eng. Sci.* **2023**, *10*, 5. [CrossRef]
24. Xiao, Y.; Yang, L.; Du, Y.; Song, Y.; Shu, C. Radial basis function-differential quadrature-based physics-informed neural network for steady incompressible flows. *Phys. Fluids* **2023**, *35*, 073607. [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.

Article

Optimizing Steering Angle Prediction in Self-Driving Vehicles Using Evolutionary Convolutional Neural Networks

Bashar Khawaldeh ^{1,*}, Antonio M. Mora ¹ and Hossam Faris ²

¹ Department of Signal Theory, Telematics and Communications, University of Granada, ETSIT and CITIC-UGR, C. Periodista Daniel Saucedo Aranda, 18014 Granada, Spain; amorag@ugr.es

² Information Technology Department, King Abdullah II School for Information Technology, The University of Jordan, Queen Rania St., Amman 11942, Jordan; hossam.faris@ju.edu.jo

* Correspondence: prof.khawaldeh@gmail.com or khawaldeh@correo.ugr.es

Abstract: The global community is awaiting the advent of a self-driving vehicle that is safe, reliable, and capable of navigating a diverse range of road conditions and terrains. This requires a lot of research, study, and optimization. Thus, this work focused on implementing, training, and optimizing a convolutional neural network (CNN) model, aiming to predict the steering angle during driving (one of the main issues). The considered dataset comprises images collected inside a car-driving simulator and further processed for augmentation and removal of unimportant details. In addition, an innovative data-balancing process was previously performed. A CNN model was trained with the dataset, conducting a comparison between several different standard optimizers. Moreover, evolutionary optimization was applied to optimize the model's weights as well as the optimizers themselves. Several experiments were performed considering different approaches of genetic algorithms (GAs) along with other optimizers from the state of the art. The obtained results demonstrate that the GA is an effective optimization tool for this problem.

Keywords: autonomous driving; artificial intelligence; convolutional neural network; evolutionary computing; genetic algorithm; machine learning; deep learning

1. Introduction

Driving is a daily activity in which a person faces many challenges and situations, including environmental and road conditions, other drivers' behavior, and road signs. Thus, self-driving is a challenging, commonplace topic nowadays and a subject of research with many variables. The world, especially through the use of artificial intelligence (AI) techniques, is moving toward a rapid revolution in the development of robotic and self-driving systems [1].

A self-driving system must possess the capacity to adhere to a predefined route and to comply with established traffic regulations. Furthermore, it must be capable of navigating the surrounding traffic in a manner that is both safe for and considerate to other road users in addition to adhering to the established regulations. Roads are not normally straight; there can be curves and meanders along the way. Moreover, there could be different situations: a single-lane road in one direction, a two-lane road in two different directions, or a multi-lane road in two different directions, for instance.

Based on the literature [2–4], analysis, and practical applications, it is evident that controlling a vehicle's steering angle is crucial for ensuring its safety and performance. The ability to control a vehicle's direction significantly impacts its trajectory, stability, and responsiveness, whether navigating sharp turns on winding roads or facing challenges in tight urban environments. The primary factor to consider, regardless of the context, is the steering angle, which determines the vehicle's movement.

The vehicle is upright when it aligns completely with the axis of movement. Moving the steering wheel causes the vehicle to deviate either right or left by applying force to the

appropriate wheels, resulting in a change in the angle between the two axles. The car is directed in the desired direction based on this angle.

Choosing the optimal steering angle is a major challenge for the driver due to the importance of these angles as well as the challenges the driver may face in making appropriate decisions. Therefore, the focus of this work was to develop methods for accurately predicting the values of these angles and guiding the driver accordingly.

Developing a self-driving system requires a lot of effort, money, and risk. Therefore, the best approach is to start working and researching in a simulator and then transfer that experience to the real world. Several simulators were considered and evaluated, and finally, the Udacity simulator [5] was chosen for its ‘simplicity’, its precision, and the fact that it allows the collection of datasets while ‘online’ (i.e., during simulation). These can either be sets of images or raw information from the car, such as the steering angle, speed, camera position, etc. These values can be used to label data.

There have been many studies on the development of self-driving systems. Most of them have applied artificial neural networks (ANNs) to build and train a model [2,3,6–8]. In this study, we also considered an ANN approach, namely, a deep learning variation called a convolutional neural network (CNN) [9] in other domains; it can be also utilized in this problem, since we considered images for defining the dataset to be used.

In addition, a type of evolutionary algorithm (EA) [10,11] was applied to optimize the CNN configuration and performance. EAs are metaheuristics derived from Darwin’s theory, which are based on the use of a population of potential solutions to a problem that are considered as individuals. These are each assigned a fitness value (depending on their quality), and then the best is normally (not always) combined with other solutions aiming to obtain better individuals in the next iteration (called a generation). EAs are very effective for solving optimization problems [12] and can also be used to solve complex problems [10]. These algorithms have shown remarkable effectiveness in tuning weights and probabilistic parameters, for instance, in gaming environments [13]. Thus, it would be interesting to explore their potential use in other contexts, such as improving CNN models in the autonomous driving context. Specifically, in this paper, the genetic algorithm (GA) [14] was applied to optimize the CNN models since it is one of the most widely used algorithms in the literature [15]. In addition, it is a pretty simple approach which has been considered in this preliminary study, making it easier (than using more complex algorithms) to analyze its potential impact or influence in the CNN’s performance.

This combination (CNN + GA) was used to predict the optimal steering angle value of a self-driving car. Thus, this study includes the application of several different CNN approaches to predict the aforementioned angle for a self-driving system. The objective is to improve the performance and accuracy of the deep learning models by optimizing the weights they depend on.

Different operators and configurations of evolutionary algorithms were tested in order to find the best set of parameters to optimize the weights of the CNN models.

Moreover, several experiments were conducted, applying the GA to different variations of CNNs, in addition to other approaches using several state-of-the-art optimizers to improve the model performance. Then, all the optimized models were compared to see the impact of the optimization (GA) and to select the best overall model.

Given that a CNN is an algorithm to be applied on images, we composed a dataset using an equipped vehicle that recorded video while driving and stored it as images divided into samples. Each sample consisted of three images taken from different camera positions (right, center, and left) and labeled with the steering angle. This was conducted inside the Udacity simulator, which allows the manipulation of steering angle parameters. It also permits placing three cameras in the desired positions on the car, and then it records a video while the vehicle is moving on the simulated road. A series of images is captured within each frame, accompanied by supplementary data pertaining to the vehicle, including the steering angle.

This dataset, which is part of the contribution of this work, has been lately refined; we applied both image processing methods (to extract the most relevant information from each image), and data balancing [16] (namely, an undersampling approach). The dataset has been improved through the inclusion of thousands of additional labeled images aiming to cover many more driving situations in the training circuit than other existing approaches in this domain.

The rest of this paper is structured as follows: The following section presents the *state of the art* of some previous related studies and their results. The third section, *Materials and Methods*, describes the methodology used to develop our proposed autonomous driving model to predict the steering angle in a driving system. So, it describes the dataset extraction and preprocessing, the CNN model structure, and the proposed optimization methodology. Next, the *Experiments, Results, and Discussion* section presents the experiments conducted and the obtained results, as well as a deep analysis and comparison of the different approaches tested. Finally, the last section comments on the conclusions reached and includes some insights for future work.

2. State of the Art

The need for operational autonomous vehicles has motivated growing research to find solutions and algorithms that optimize various aspects of self-driving. The ultimate goal is to develop a fully integrated autonomous driving system.

Most of the studies have been conducted using vehicle driving simulators. Thus, there is research that has developed domain randomization to transfer from simulation to the real world based on deep neural networks [17], but in 2020, a group of researchers tried to train this type of neural network in CARLA (Car Learning to Act), an open-source simulator for autonomous driving research [18,19]. It did not work well when trained individually; it could not be transferred to the real world, so viewpoint augmentation (a popular preprocessing trick to improve generalization accuracy) was applied to it. It was called DR-AUG (Domain Randomization–Augmentation) [20]. In the same line, a research technique was also used to train an autonomous driving model on a simulator without using labels (objects with information) from the real world [21].

There are many simulators, including VISTA (Virtual Image Synthesis and Transformation for Autonomy), which is a data-driven simulator for end-to-end autonomous driving training [20]. However, maybe the most used simulator is Udacity [5], in which several different approaches have been tested, mainly convolutional neural network (CNN) models for autonomous driving [2,3,6,7].

The CNN training is performed on data, which could be a set of images collected while driving on different types of roads, residential streets (paved or unpaved), or highways, in different weather conditions. In one of the previous works [22], they collected data over 72 hours of driving in many different conditions in the real world and captured by one camera mounted on the center of the windshield of the car (later complemented with some ‘transformed’ images). These images were then labeled with some extra information about the car status. The CNN model is able to learn efficiently to drive in many different situations, but it seems to be not very robust (not completely reliable in all the situations).

Our study follows a similar methodology, but that dataset was not published. Moreover, since we used the Udacity simulator to collect the images, the precision of our samples is likely higher regarding the rest of the information about the car (the steering wheel, in our case). Moreover, we tested the CNN model with different optimization methods, aiming to obtain a definitive (and reliable) approach. Finally, study [22] used very powerful (and expensive) computational resources, while our approach was run on a much humbler setup.

Other studies have used data from simulation and translated them to the real world [21]. The authors also applied deep learning approaches based on images, obtaining a feasible solution (as a proof of concept).

In the work [23], the authors created a ‘simple’ (smartphone-size) physical model of a car and applied different realistic scenarios where it should drive autonomously. They also used supervised training with a CNN as well as an implementation of deep Q-learning to train the car on recognizing some road signs.

A CNN was also used for object detection and lane tracking for autonomous driving using a real dataset collected in the Rancho Palos Verdes area around San Pedro [24].

Another proposal [4] used CNN, Long Short-Term Memory (LSTM), and Fully Connected (FC) layers to control the steering angle, also considering data gathered from Udacity, using two vehicles capturing images from different perspectives and combining them. In the present work, the Udacity simulator was also used.

Following other proposals, in this paper, a CNN model for autonomous driving was trained using image data samples. However, in our case, the samples were labeled with an accurate steering angle. Each sample consists of three images collected from three camera positions (right, center, and left) by driving the car in the Udacity simulator, recording a video, and then saving the data as images.

Furthermore, our study was conducted using a genetic algorithm to select the best learning scheme, which includes preprocessing, attribute selection, and learning algorithms. The data were analyzed, and a set of potential solutions was generated to evaluate their performance and effectiveness in achieving the identified learning objectives [25].

This research aims to develop convolutional neural network (CNN) models and justify the weights used in them to achieve maximum accuracy in predicting steering angles in autonomous driving systems. Although many studies have preceded us in using artificial neural networks to build and train models [2,3,6,7], current research has not only been limited to creating an effective predictive model for steering angles but also focused on combining a convolutional neural network (CNN) and genetic algorithms (GAs) to improve model performance in steering prediction. In addition, efforts have been made to develop and train a CNN model using our own data extracted from the Udacity simulator.

The present study represents an advancement over previous research in the domain of autonomous driving utilizing a convolutional neural network (CNN) in two main aspects: the generation of a more reliable and complete dataset and the application of evolutionary optimization techniques over the CNN models.

While numerous previous studies have concentrated on the utilization of simulation data or genuine data for the training of autonomous driving models, our approach has firstly focused on the generation of a much more complete (and accurately labeled) image-based dataset. Using the Udacity simulator tool to gather images from the car’s perspective, we conducted a process of systematic experimentation and expert revision for the generation of thousands of images representing a wide variety of driving situations. Thus, this led us to complete a dataset from which the models can be better trained, being then more reliable and effective in the prediction of steering angles.

Moreover, our study is distinguished by its integration of genetic algorithms with CNNs, with the objective of improving their performance. While numerous previous works have primarily focused on the creation of ‘pure’ neural network models using standard optimizers, our research aims to provide a comprehensive comparison of the performance of the CNN model using a wide variety of optimizers, such as Nadam, Adam, RMSprop, Adamax, SGD, Adagrad, Adadelat, and FTRL, with the CNN + GA model (using a GA as the optimizer). In addition, the application of GAs as an enhancing tool to improve the performance of the best optimizers was also studied in this work.

3. Materials and Methods

This section introduces the considered simulator and describes the collected dataset (with the applied preprocessing and enhancing methods) as well as the deep learning model that was applied.

3.1. Udacity Simulator

The Udacity simulator [5] is an open-source self-driving car framework developed with the Unity Engine. It is available at <https://github.com/udacity/self-driving-car-sim> (accessed on 5 June 2023).

Udacity is designed for students and researchers, since it offers several utilities to train self-driving car models, mainly using deep learning techniques, which can be integrated in the framework. Among its tools, the possibility of recording the driving session is offered in order to create your own dataset to be later processed.

Moreover, it is not very demanding with regard to computing resources, so it does not require a high-level machine. Finally, Udacity is quite easy to use, and it also achieves the desired goals of precision and accuracy, since the simulation is very realistic in several aspects.

Figure 1 shows a screenshot of the Udacity simulator, where its realistic graphics appearance can be seen.



Figure 1. Udacity simulator.

It allows video to be captured at 24 frames per second (fps) from three cameras (right, center, and left) and later stores it as a set of JPG images. It also allows storing a text file containing additional information, such as image title, steering angle, throttle, reverse, and speed, among others.

This simulator provides two driving tracks (shown in Figure 2, right side): the Lake Track spans approximately 1.1 km of a quite 'simple' profile. However, it contains standard curves in both directions, which will be very useful for the controller to 'learn' driving. The other circuit is named the Jungle Track, with a length of 1.5 km and a much more complex profile (see Figure 2, left side) with a lot of curves.

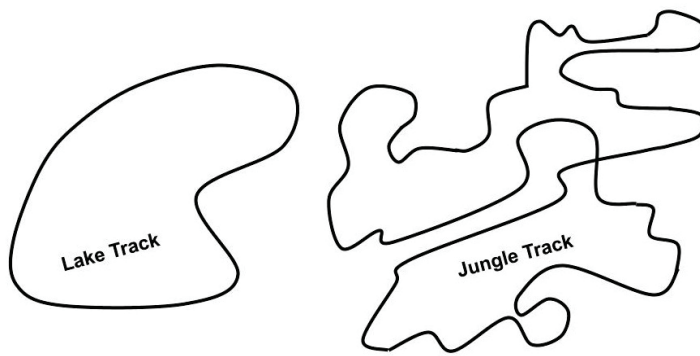


Figure 2. Udacity's Lake and Jungle Tracks.

The Lake Track was chosen as the training circuit for the model because of its diverse profile, which provides varied driving data for recording and training. Additionally, the track offers a simpler circuit for initial model training.

The Jungle Track was considered for verification and evaluation purposes. This is due to the circuit's complexity and challenges. This approach ensures a balance between diversity and result validation, ultimately improving the final model's quality and generalizability to various conditions and regions and, furthermore, ensuring the model's adaptability to real-world conditions and validating the results.

3.2. Dataset and Processing

Data for self-driving systems are normally very specific for a proposed model or even private [22]. For this reason, we decided to compose our own dataset. Following other approaches, we aimed to use an image-based dataset recorded from the car perspective. However, instead of just considering one frontal image, we placed three different cameras (with three different angles) on board the simulated car. This added a richer diversity of 'views' to the dataset. Figure 3 shows an example image from each camera angle.

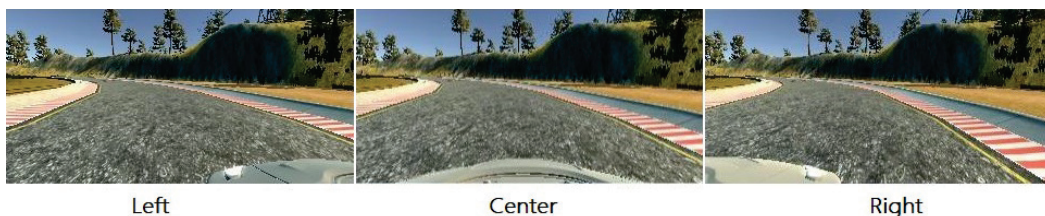


Figure 3. Example of images of three camera positions.

The dataset was collected while a human expert controlled the car driving in the Udacity simulator. A 25 min recording session on the Lake Track was conducted utilizing three vehicle-mounted cameras located on the left, center, and right sides. The cameras captured an assortment of driving scenarios, such as varied curve entry speeds and steering angles. The recording was captured at 24 frames per second, and the average vehicle speed during the recording was 7.2 mph.

Thus, each sample in the dataset is composed of three images and the associated information at that specific moment in the simulator, such as the steering angle or the current speed [26].

Once the dataset was created, we conducted a preprocessing phase. Then, an undersampling process [27] was applied: firstly, identical images with the same steering angle were removed, since they do not add relevant information for the training of the self-driving model. Many of these images were stored in the recordings, since 24 frames are saved every second. Secondly, there was a large amount of samples with a steering angle equal to 0 (which means 'do not take any action or move straight ahead'), so these

samples were removed. After undersampling, the data distribution became more logical, as illustrated in Figure 4, which shows the distribution of the original dataset.

There are several justifications for the exclusion of samples with a steering angle of 0 (i.e., straight-line driving) from prediction models, with regard to both methodology and theory. Primarily, the exclusion of a considerable number of straight-line driving samples allows for the creation of a more balanced dataset, which is essential for the development of an accurate model. The incorporation of a substantial number of straight-line samples may result in the development of a model that is excessively simplistic, which would subsequently diminish its accuracy in anticipating curves or complex scenarios. Secondly, focusing on disparate steering angles enhances the model's predictive capacity, as it responds to data that necessitate a more intricate examination. Thirdly, this methodology minimizes noise and redundancy within the data, thereby enhancing the model's efficacy in processing diverse scenarios and optimizing the quality of the predictions. This, in turn, renders the model more balanced and effective in handling disparate driving scenarios.

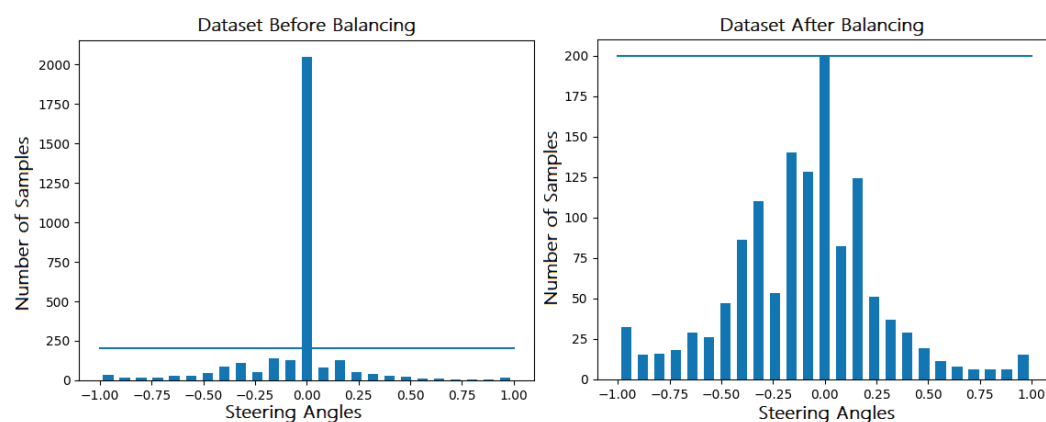


Figure 4. Distribution of dataset before and after data undersampling.

At the end of this process, there were 3882 images left, each of which was taken from different positions (left, right, or center). In order to use the data to train the controllers, the original dataset was divided into 80% of samples for training and 20% for validation. These samples contain a wide variety of steering angles, as illustrated in Figure 5.

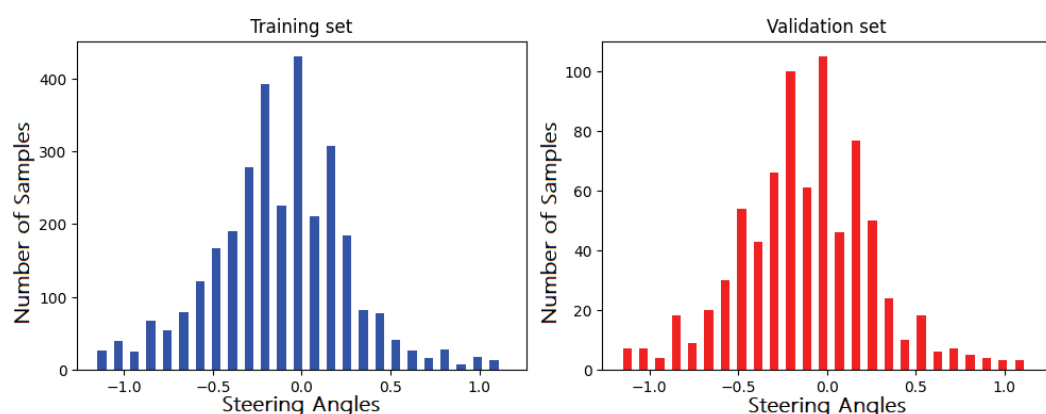


Figure 5. Original training and validation dataset distribution.

However, the original dataset was insufficient for training a model capable of handling all test conditions effectively. As a result, additional data were carefully added (by an expert) after thoroughly assessing the dataset's limitations, yielding a novel dataset with a total duration of approximately 61 min. Thus, it included 35,854 new images covering a much wider number of driving situations (steering angles with low representation in the

original dataset). The data recording session was conducted on the Lake Track with the use of three cameras positioned on the vehicle's left, middle, and right sides, respectively. The cameras documented a variety of driving scenarios, including different corner entry speeds and steering angles, with the recording captured at 24 frames per second. Following again the undersampling process, a total of 11,658 usable images were obtained. Furthermore, the novel dataset was divided into 80% of samples for training and 20% for validation. These samples contain a wide variety of steering angles, as illustrated in Figure 6.

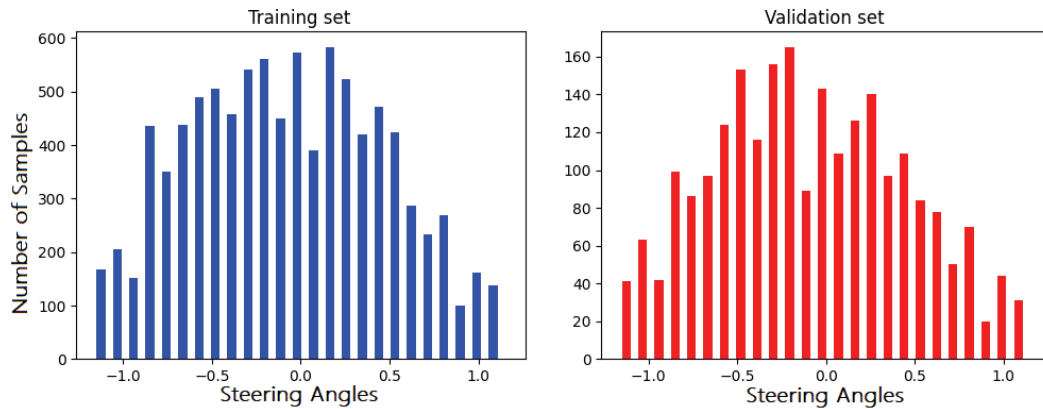


Figure 6. Novel training and validation dataset distribution.

Later, the dataset was augmented, aiming to create new images with new properties by changing some of the existing similar ones. This process helps to overcome underfitting, which occurs when a model is too simple to capture the underlying patterns in the data, especially when there is insufficient training data. This was performed using image augmentation functions (imgaug and cv2 library) with randomly selected images from the dataset. Refer to the examples depicted in Figure 7. Namely, the applied processes were as follows:

- **Zooming in:** enlarging the dimensions by 130% on both the x and y axes to remove non-relevant information of the screen borders in order to enhance the extraction of features.
- **Brightness change:** multiplying all pixels by a random number in the range (0.2, 1.2); thus the image becomes darker if the random number is less than 1, and if the number is bigger than 1, it becomes brighter. This allows the model to extract image features across varying luminosity.
- **Panning shift:** moving from -10 to +10 pixels on each axis: x and y. This gives some variation to the images, aiming for the model to concentrate on the most critical aspects of the image.
- **Flipping:** mirroring the image horizontally and also changing the label of the steering angle to its complementary value (negative if it was positive and the other way round). This increases the chance of the model to extract features from the images in the opposite direction with a negative steering angle.

After the augmentation process is complete, an example of this process can be observed in Figure 8, as well as in other works that utilize images for deep learning models [6]. All the images in the dataset were processed to remove unimportant details. This helps to speed up and refine the training stage. Thus, a combination of operations was conducted applying the CV2 library: (1) Cropping the image (to ensure the model focusing on the most important features in the image, such as road lines and road edges). (2) Converting the image from RGB to YUV, since YUV reduces the bandwidth in comparison to RGB, resulting in higher efficiency. (3) Using a Gaussian blur filter to smooth the image. (4) Resizing the image to 200 × 66 in order to better fit with the used CNN architecture (described in next subsection).

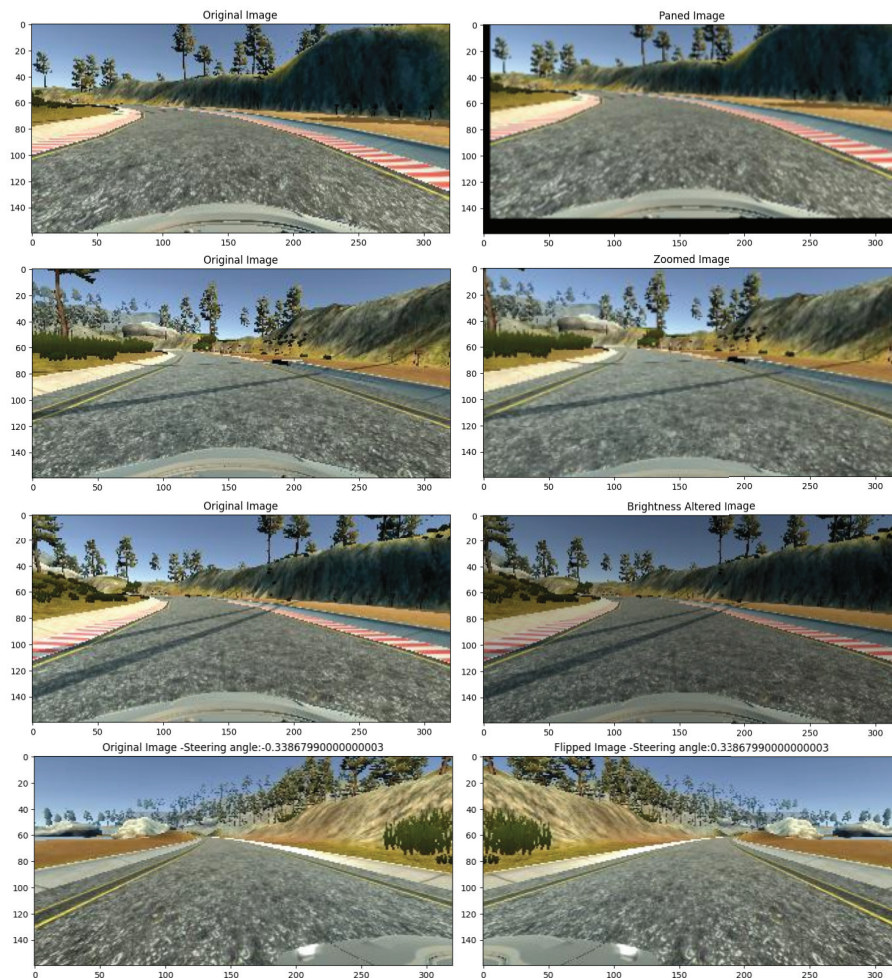


Figure 7. Augmentation examples.

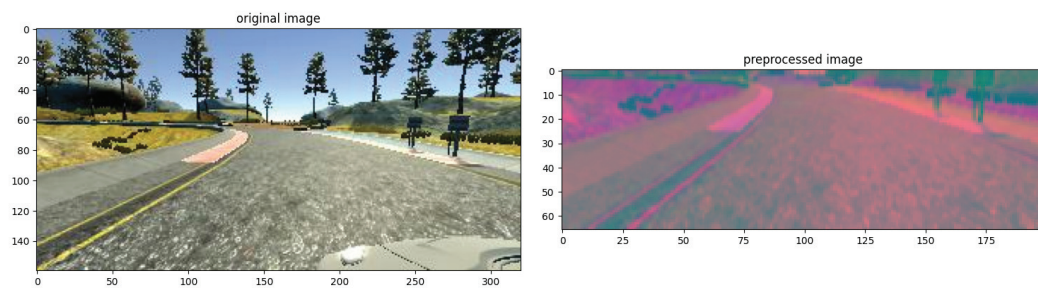


Figure 8. Original image and image preprocessed by cropping, RGB to YUV, Gaussian blur filter, and resizing.

3.3. CNN Model Architecture

A convolutional neural network (CNN) is a deep learning (DL) method designed to process images.

Prior researchers have achieved considerable success in employing this deep learning (DL) technique for pattern recognition in video [26]. Furthermore, the extraction of more useful and powerful features is feasible through the exploitation of advances in machine learning, particularly those pertaining to deep learning [28]. Another study focused on plant disease classification, and a convolutional neural network (CNN) was applied over the features extracted from plant leaf images. This convolution process is summarized in the inputs, filters, and outputs, which include multiple layers and neurons [29].

In this study, the NVIDIA architecture [22] was considered to build the CNN based on previous research findings [2,6].

In a previous study [6], the NVIDIA architecture was compared to a less complex structure. NVIDIA demonstrated superior simulation performance, achieving an accuracy rate of 96.83% compared to the other architecture's accuracy rate of 76.67%. These results suggest that the NVIDIA CNN architecture outperforms other architectures in terms of performance and accuracy, making it an attractive option, particularly in areas such as artificial intelligence and efficient image processing.

The considered configuration for the CNN utilizes the parameters defined in different works of the literature [6,22]. Thus, the kernel size is set to 5 by 2 in the first three layers and 3 by 3 in the following two. Figure 9 shows the considered CNN architecture, which uses images extracted from the simulator.

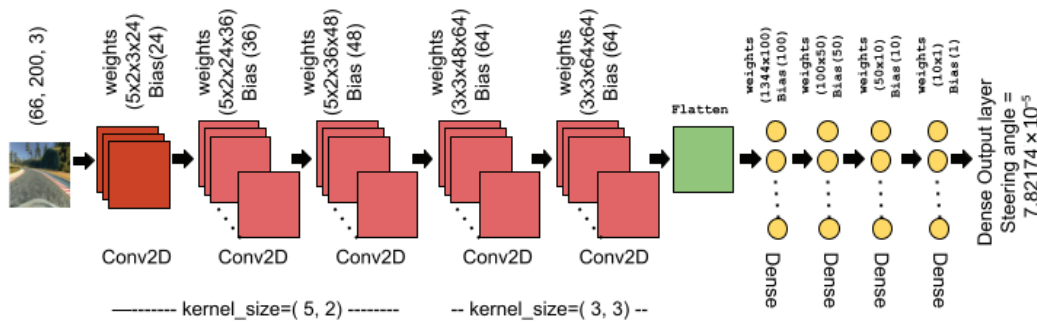


Figure 9. Convolutional neural network (CNN) for steering angle prediction based on NVIDIA architecture.

As can be seen in Figure 9, the architecture consists of eleven layers; the first is the input layer containing the car trajectory samples (images). Each sample is labeled with a steering angle value in the range $[-1, 1]$, where a negative angle indicates a left turn and a positive angle a right turn. This is followed by five layers with different filters. Then comes a layer to flatten all the inputs, followed by four dense layers. Finally, the output layer yields the predicted steering angle [6,22].

An exponential linear unit (ELU) was used as the activation function in all the layers, since ELUs accelerate learning in deep neural networks and lead to reach a higher accuracy. An ELU also improves learning compared to units with other activation functions [30].

3.4. Evolutionary Optimization by Means of Genetic Algorithm

Evolutionary algorithms (EAs) are inspired by Darwin's theory in which individuals adapt to survive and eliminate traits that make them weaker [10].

EAs have many approaches, such as evolutionary programming, evolutionary strategies, genetic programming, and genetic algorithms [10]. In this study, the genetic algorithm (GA) was chosen because it is one of the most applied algorithms for optimization in the literature [12,15], and it is quite simple to implement and combine with other approaches. The GA displays clear maps of the natural evolutionary process on the computer, where it is used in machine learning, pattern recognition, and optimization problems [10].

The genetic algorithm (GA) is a population-based method, in which every individual (or chromosome) is a possible solution for the optimization problem to solve. An individual is composed of a set of values that are called 'genes'. The GA applies four main operators:

- Selection: a set of individuals (normally the best) is chosen to be the parents of the following generation (iteration).
- Crossover: a function that combines the material of two individuals (parents) to create some descendants (a mixture of the combined ones).
- Mutation: a random variation in the individuals' genes.

- Replacement: a substitution of some individuals (normally the worst) by the new generated ones.

A GA strongly depends on the proper definition of an Evaluation Function, normally denoted as a fitness function.

In the context of evolutionary algorithms, each solution is typically evaluated by calculating its fitness value, which facilitates the subsequent selection process. This allows the algorithm to identify the solution that most closely approximates the desired outcome. In the context of our study, we employed a generational genetic algorithm [31], whereby the selected solutions were subjected to a process of mutual recombination. This operator enables the exchange of characteristics between two or more parental solutions, thereby producing new offspring. The objective of this approach is to facilitate the transmission of beneficial traits, thereby enhancing the quality of the population in subsequent generations.

It is important to note, however, that not all evolutionary algorithms rely on combination techniques. To illustrate, a steady-state algorithm [32] may prioritize selection and replacement strategies, thereby circumventing the necessity for a crossover process. Moreover, there are sophisticated evolutionary algorithms, such as CMA-ES [33], xNES [34], and OpenAI-ES [35], which prioritize the enhancement of a single individual rather than the population as a whole. These algorithms employ techniques such as self-adaptation and evolutionary strategies. To provide a comprehensive overview of evolutionary algorithms, we expand our discussion to include these alternative approaches, providing relevant citations to enhance our understanding of evolutionary algorithms.

We considered the GA for the so-called metaoptimization of the CNN model. So, the GA was applied to optimize the weights of the model. The algorithm follows the loop presented in Figure 10.

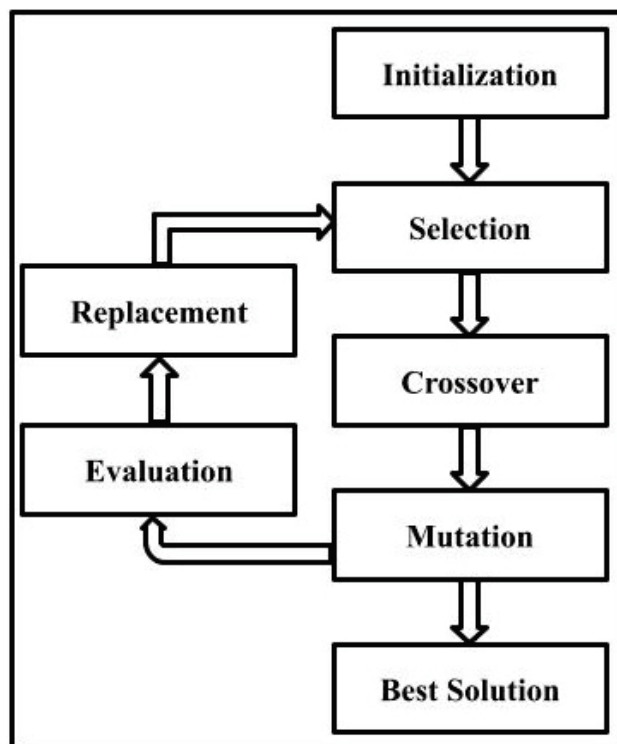


Figure 10. Genetic algorithm procedure.

Thus, in our approach, CNN models were considered as individuals, and their weights were the genes. A generational approach was considered, so half of the population are selected as the parents of the following generation. Two different selection operators were used: best individuals of the population and random selection.

Regarding the crossover operation, a series of weights in a layer is transferred from one model/individual to another, and the layer is chosen randomly. As Figure 11 shows, a sequence of 24 weights in one layer of the model is transferred to the same layer of the other model, with the position of the sequence in both parents chosen randomly.

For the mutation, the weights within the model are modified in a random manner. This is performed with the intention of enhancing diversity and exploring new solutions. This is achieved by replacing the specified weight with a random value drawn from a uniform distribution between 0 and 1.

This research examined two types of population replacement, either replacing completely the previous population with the best new individuals or replacing the population with a mix of the selected parents and new children.

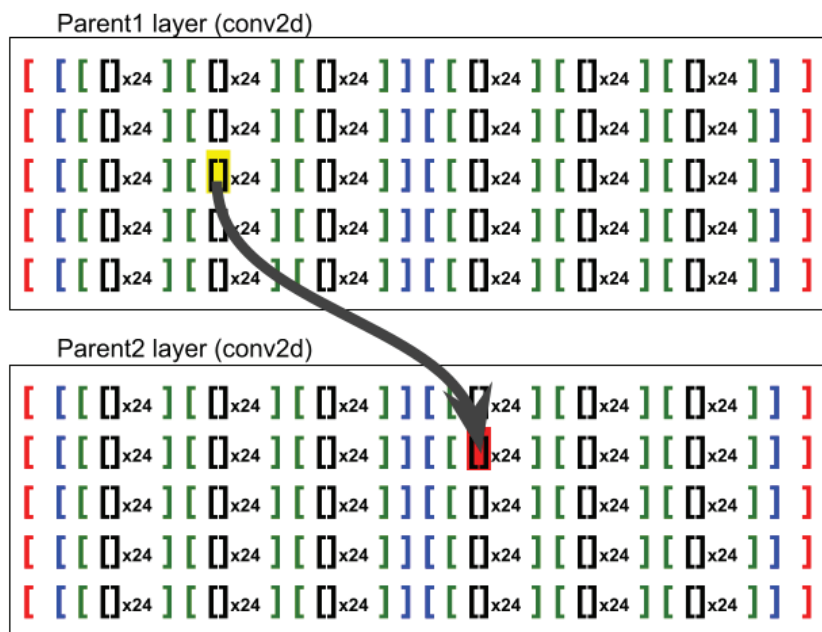


Figure 11. Example of crossover process.

Algorithm 1 also describes the GA process. The algorithm is based on a genetic algorithm, which is used to optimize a population of individuals with the objective of minimizing the loss function and achieving enhanced performance. The algorithm commences with the delineation of a set of parameters, including the mutation probability (p_m) and a generation comprising 50% of the population ($n = 16$) with an initial population of 32 individuals (L). The process commences with the iteration of several generations (MAX_GENS), during which the parents are selected either based on the least loss or randomly to generate new individuals. Crossings are performed between individuals to generate new offspring, and mutations are performed on the resulting offspring according to the mutation probability in order to enhance diversity and explore new solutions. Subsequently, the new individuals are evaluated, and the least efficient individuals in the population are replaced in accordance with the specified scenarios. This may be accomplished by comparing the fitness of the new individuals ($Indivi_i2$) with that of existing individuals or by combining the performance of the parents with that of the offspring. The objective is to optimize the performance of the population over generations. This is accomplished through the implementation of selection, crossing, and mutation, which collectively facilitate the attainment of the optimal solution with minimal loss and maximal efficiency.

Algorithm 1: Genetic algorithm optimizer.

Parameters: Mutation probability $P_{mut} = 0.2$ (or 1)
Parameters: Population size $N = 32$ (16 parents per generation)
Parameters: Generations MAX_GENS
Input: Initial population of individuals $P = \{Ind_1, Ind_2, \dots, Ind_N\}$
Output: Optimized population

```

1 for  $g = 1$  to  $MAX\_GENS$  do
    // Selection of Parents
2    $ParentPool \leftarrow []$ 
3   for  $i = 1$  to  $\frac{N}{2}$  do
4     if Selection criterion (e.g., minimum loss or random) then
5       |  $ParentPool.append(selected\_parent)$ 
6     end
7   end
    // Crossover and Mutation to Generate Offspring
8    $Offspring \leftarrow []$ 
9   for  $i = 1$  to  $\frac{N}{2}$  do
10     $Parent_1, Parent_2 \leftarrow$  Randomly select two parents from  $ParentPool$ 
11     $Child \leftarrow Crossover(Parent_1, Parent_2)$ 
12    for  $j = 1$  to  $genes\_count$  do
13       $r \leftarrow$  random number between  $[0, 1]$ 
14      if  $r < P_{mut}$  then
15        | Mutate gene  $j$  of  $Child$ 
16      end
17    end
18     $Offspring.append(Child)$ 
19  end
    // Replacement Strategy
20  Evaluate fitness of all individuals in  $P$  and  $Offspring$ 
21  foreach individual child in  $Offspring$  do
22     $parent\_fitness \leftarrow$  fitness of corresponding parent
23     $child\_fitness \leftarrow$  fitness of child
    // Scenario 1: Replace Parent if Offspring is Better
24    if  $child\_fitness < parent\_fitness$  then
25      | Replace parent with child in population  $P$ 
26    end
    // Scenario 2: Combine Parent and Offspring Fitness
27     $combined\_fitness \leftarrow parent\_fitness + child\_fitness$ 
28    Use combined fitness to adjust the probability of replacement
29    if  $combined\_fitness$  meets threshold then
30      | Replace parent with child
31    end
32  end
33 end
34 return Optimized population  $P$ 

```

4. Experiments, Results, and Discussion

Three experiments were conducted in this study. Firstly, we trained different approaches of convolutional neural network (CNN) models combined with different optimizers from the literature, which were analyzed. Secondly, a GA was tested as an optimizer for a CNN, evaluating and comparing eight different configurations. Finally, the GA was also applied on a fully trained model in order to fine-tune its already optimized weights (with a standard optimizer). All the results of these models were compared. In the first

two experiments, the original dataset was considered, while the third experiment was conducted also using the novel dataset to compare its influence on the results.

The Python programming language and Google Collaboratory (one machine and one GPU) were utilized to write and compile the code, with various libraries such as KERAS, TensorFlow, Scikit-learn, Pandas, NumPy, CV2, and Imgaug integrated for added functionality.

4.1. Experimental Setup

The models were evaluated considering the mean squared error (MSE) as a loss function. It calculates the mean of the squared difference between observed and predicted values [36]. See the formula in Equation (1):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (1)$$

where n is the number of observations/samples, y_i is the value of the i -th observation, and y'_i is the i -th predicted value. The learning rate was set to the default value in the CNNs.

The GA was executed for 50 generations (iterations) with a population of 32 individuals (models), using the aforementioned selection, crossover, mutation, and replacement approaches. Fitness was calculated by determining the mean squared error (MSE) for each individual when running the CNN model with its respective weights.

A total of 30 runs were performed for each non-deterministic optimizer and 10 for each GA approach (due to the high demand on computing time of this algorithm).

4.2. Training CNN Model Using Different Optimizers

In the first experiment, eight existing optimizers were studied, namely, Nadam, Adam, RMSprop, Adamax, SGD, Adagrad, Adadelta, and Ftrl [37].

The *loss value* was calculated using the mean square error (MSE) on the training data (TD) and also on the validation or test data (VD). The average loss for the 30 models trained by each optimizer is shown in Figure 12.

Table 1 also presents the results but in a numerical way. Training the models by using the Nadam optimizer had the lowest average loss based on the training dataset, while the highest mean loss was obtained using Ftrl. In addition, RMSprop and Adam obtained a very good result, as RMSprop had the lowest loss based on the validation dataset. Therefore, we used the models trained with this optimizer in the third set of experiments, which is explained in Section 4.4.

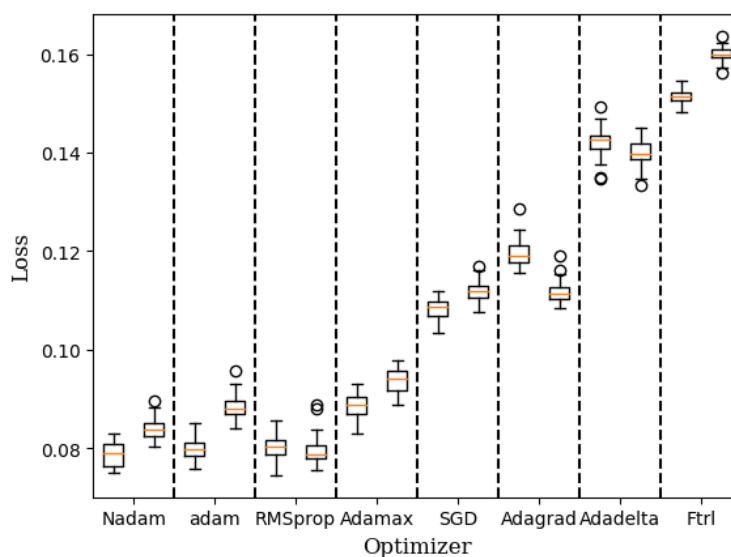


Figure 12. Loss (MSE) boxplots for the 30 trained models for each optimizer, based on the training dataset (left boxplot) and validation dataset (right boxplot).

Table 1. Loss (MSE) results: minimum, average (Avg), and standard deviation (SD) of 30 runs on the training dataset (TD) and the validation dataset (VD) for model training using different optimizers.

Optimizer	Data	Min Loss	Avg	SD
Nadam	TD	0.0750	0.0789	± 0.0024
	VD	0.0802	0.0838	± 0.0022
Adam	TD	0.0758	0.0799	± 0.0022
	VD	0.0839	0.0884	± 0.0024
RMSprop	TD	0.0745	0.0802	± 0.0025
	VD	0.0755	0.0796	± 0.0030
Adamax	TD	0.0829	0.0885	± 0.0026
	VD	0.0888	0.0937	± 0.0026
SGD	TD	0.1033	0.1081	± 0.0019
	VD	0.1075	0.1117	± 0.0019
Adagrad	TD	0.1155	0.1195	± 0.0026
	VD	0.1084	0.1116	± 0.0022
Adadelat	TD	0.1346	0.1422	± 0.0030
	VD	0.1332	0.1398	± 0.0028
Ftrl	TD	0.1480	0.1515	± 0.0015
	VD	0.1562	0.1600	± 0.0015

4.3. Optimizing CNN Model by Means of GA Using Different Configurations

As stated before, in the second experiment, an optimization method based on a GA was implemented and applied to a CNN to train a self-driving model. The considered evaluation function (fitness) was the loss value, since a smaller loss means a smaller error and therefore a better optimization result.

This section also studies the impact of the different configurations for the genetic operators (selection, mutation, and replacement), as described in Section 3.4.

Thus, 10 runs were performed with each of the following configurations:

- BBH: selects the best individuals, replaces the population with the best individuals, and has high mutation probability.
- BBL: selects best individuals, replaces population with best individuals, and has low mutation probability.
- BNH: selects the best individuals, replaces the population with parents and children, and has high mutation probability.
- BNL: selects the best individuals, replaces the population with the parents and children, and has low mutation probability.
- RBH: randomly selects individuals, replaces the population with the best individual, and has high mutation probability.
- RBL: randomly selects individuals, replaces the population with the best individual, and has low mutation probability.
- RNH: randomly selects individuals, replaces population with parents and children, and has high mutation probability.
- RNL: randomly selects individuals, replaces population with parents and children, and has low mutation probability.

Tuning a model such as a CNN with a huge amount of weights would be an extremely demanding task if we aimed to test all the possibilities, given that there could be millions of

possible combinations. Thus, applying a GA can perform this task in a reasonable amount of time.

In this study, the GA optimizer starts from a random distribution of weights in the initial population, and it is able to evolve to a competitive configuration for the CNN, as shown in the results presented in Table 2 and Figure 13 after 50 generations. To enhance clarity and effectively emphasize the performance of the optimal optimizer, we decided to limit the range of the curves (graphs) presented in our analysis, since showing the complete ranges would have made it harder to see the essential details necessary for a clear understanding of the key findings. In particular, the RNH and RNL results are cut, because these approaches are based on random selection methods, which lead to considerable variability in the results.

Table 2. Loss (MSE) results of 10 runs applying GA on the CNN model with different configurations for selection function, replacement policy, and mutation operator. Minimum, average (Avg), and standard deviation (SD) are included.

	Min Loss	Avg	SD
BBH	0.1445	0.1471	± 0.0016
BBL	0.1417	0.1451	± 0.0020
BNH	0.1445	0.1472	± 0.0019
BNL	0.1433	0.1459	± 0.0015
RBH	0.1443	0.1472	± 0.0013
RBL	0.1436	0.1457	± 0.0016
RNH	0.4200	8.4456	± 7.1396
RNL	0.1498	0.2496	± 0.2076

As can be seen in the table and figure, all the approaches yielded similar results, excepting RNH and RNL, which are the configurations that are more focused on exploration, i.e., they add more random components to the search than the rest of the approaches: random selection of parents and replacement with children even if they are worse than previous individuals in the population. This can be also seen in the high value of the standard deviation in both cases.

The best results were obtained by the BBL configuration, which is the approach that most focuses on exploitation, i.e., selecting the best parents overall, replacing with the best individuals, and applying a low mutation factor.

In comparing the optimizer based on a GA and the rest of the optimizers from the literature, we see the best solution (0.1417) was obtained by BBL. It is better than the best solution obtained by the Ftrl optimizer (0.1480).

Given that this proposal is a proof of concept conducted using relatively low computational power (one machine with one GPU), we can consider these results to be promising. This also leads us to think that obtaining a better loss than the best optimizers is doable, but it would be a very demanding task using a GA, because this method is, at the end of the day, a search algorithm inside a space of possible solutions, and the number of combinations of weight values is gargantuan.

However, considering the great power of GAs, they can obtain the optimal combination of weights if we give them enough time/computational power, as they are able to move to different parts of the search space, avoiding local optima, which is something that most of the standard optimizers cannot achieve.

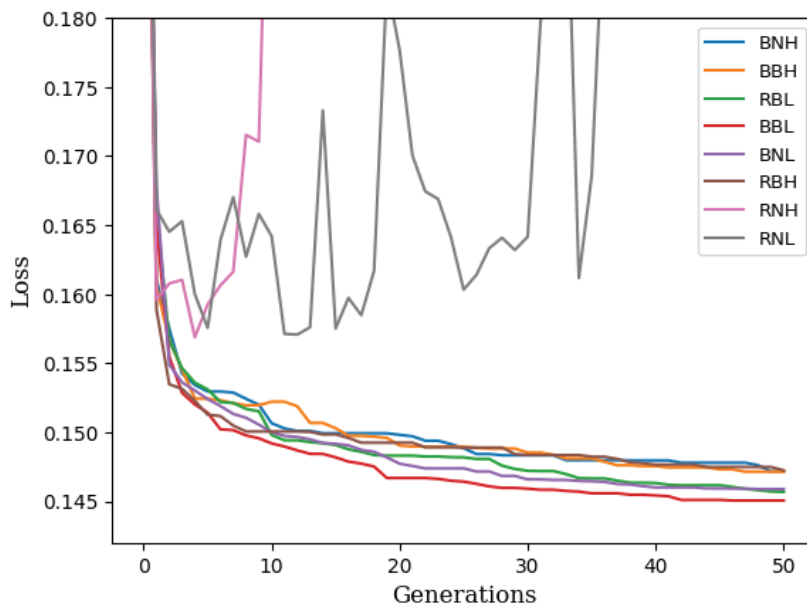


Figure 13. Average loss (MSE) of 10 runs for each of the different configurations for the GA optimizer over the CNN model.

4.4. GA Optimization of a Fully Trained (and Optimized) Model

Finally, the last experiment assessed the value of the GA for the enhancement of already optimized models. To this end, we considered the most effective approaches outlined in Section 4.2, namely, CNN models with RMSprop, Adam, or Nadam optimizers. Thereafter, a GA with the BBL configuration (the optimal outcome of the second experiment) commenced with the set of weights defined by that optimizer (of 32 models trained by RMSprop, Adam, or Nadam as population) and conducted an additional evolutionary process.

Table 3 presents a comparative analysis of three optimizers—namely, Adam & GA, Nadam & GA, and RMSprop & GA—on two distinct datasets. The original and novel datasets were employed to assess the performance of the optimizers. The results demonstrated that RMSprop & GA achieved the lowest minimum loss on the original dataset, indicating superior training efficiency. Additionally, Adam & GA exhibited consistent performance across the two datasets, achieving the lowest minimum loss on the original dataset compared to the other optimizers. Overall, RMSprop & GA demonstrated superior performance in reducing losses, particularly during training. However, in the simulation, Adam & GA was the most effective optimizer, a finding that is discussed in the subsequent section.

As can be seen, an improvement is obtained in all cases, which proves again the value of GAs. It also reinforces our hypothesis regarding the application of a GA as the optimizer in order to obtain smaller loss than the rest of optimizers.

Although the difference in the optimization is not big, it was consistent with the application of a GA for only 50 generations (which will be bigger with more generations). Moreover, we can consider it as relevant due to the sensitivity of the field in which this study is enclosed. Thus, any error in the self-driving vehicle is a potential threat to human life or to material losses. Furthermore, the aforementioned optimization yielded favorable outcomes in terms of driver performance as assessed through simulation-based evaluations.

In this research, execution speed and optimization time were not discussed because all the tested optimizers were integrated in Keras [37], so they can be run on clusters of GPUs or TPUs. Furthermore, Keras is built on the TensorFlow platform, which provides high-speed computation [38]. The tested GA is a preliminary approach, which would be improved if it could be implemented (or integrated) also in Keras.

Table 3. Comparison of enhanced optimizers (by means of GA) on the two datasets. Loss (MSE) value is presented at the beginning (initial population loss) and after the GA optimization process (min loss).

Optimizer	Dataset	Init Populat Loss	Min Loss
Adam & GA	Original	0.076	0.0752
	Novel	0.123	0.1199
Nadam & GA	Original	0.075	0.0747
	Novel	0.130	0.127
RMSprop & GA	Original	0.075	0.0709
	Novel	0.129	0.122

4.5. Autonomous Drivers Performance Tests

Performance tests for autonomous drivers represent a crucial component of the evaluation process; while theoretical outcomes may exhibit promising results, actual driving performance may diverge from these results. The drivers were subjected to testing in a simulator on two distinct tracks, the Jungle and Lake Tracks. The assumed checking distance was set to be three laps of each track.

Table 4 presents the simulation performance of models representing autonomous controllers driving through the Jungle and Lake tracks. The table provides the obtained results of the CNN models combined with the standard optimizers considered in this study, as well as those obtained after the improvement of the best optimizers by means of a GA, as stated previously in Section 4.4. The table shows the ‘theoretical’ evaluation based on the loss (MSE) value for the predicted steering angle, together with some performance metrics reached at the circuit, namely, max and average speed, covered distance, and time driving. In each case, the first row depicts the results achieved by training the model on the original dataset, while the next row shows the results obtained after training the model on the new dataset.

Looking at the results in the training track (Lake), it can be noticed that almost all the controllers successfully completed the three-lap test distance. Those that were not able to do this could not be considered as reliable controllers for autonomous driving following the presented approach. Focusing on the best ones, the fact that the loss in the new dataset was always worse could be remarkable. This could be interpreted as a poorer performance for the models; however, this happens due to the ‘harder’ dataset used. In spite of this, the controllers trained using the new dataset behaved much better (than those trained using the original one) when we tested them on the circuit. This reinforces the quality of the enhanced dataset we composed.

With regard to the results obtained on the Jungle Track (unknown for the controllers), it can be seen that, again, models trained using the new dataset obtain a much better performance on the circuit, while their loss is a bit worse. Actually, only models trained with the new dataset were able to finish the three laps on this track, since it is much more difficult than the training one (see Figure 2). Specifically, just Adam variations could reach the end of the test without experiencing any crashes or deviations from the specified path. Only “Adam” and “Adam&GA” made it to the end of the test circuit. Although “Adam” ended with a driver going off the track before getting back on, “Adam&GA” finished the track without any derailments. The rest of the controllers had incidents that caused them to avoid finishing (such as crashing out of the bounds of the lane). This performance is particularly impressive on the Jungle Track, given that the model had not undergone any pretraining and given its difficulty.

Focusing on the optimizers, the results also indicate that models trained with Adam outperform those trained with RMSprop, as measured by the loss (MSE). Vehicles trained using a CNN and RMSprop were only able to cover short distances before deviating from

the track during the simulation. Models trained with RMSprop & GA (RMSprop and genetic algorithm) outperformed those trained with RMSprop alone on both tracks.

Furthermore, the effectiveness of models trained with SGD, Adagrad, and Adadelata was limited as they could only cover short distances on the training and validation tracks before derailing. The case of Nadam is also remarkable, but in the bad sense, since the results from driving on the Jungle Track are poorer using the novel dataset than the original one.

Additionally, we could argue that the CNN model utilizing the Ftrl optimizer was not appropriate for autonomous driving as it did not produce angle changes, even when traveling only a short distance before veering off the track.

The results of this study offer insight into model performance, including adaptability to various environments and the impact of algorithms on final outcomes. Based on these results, it can be concluded that the models trained using Adam with genetic algorithms were the most effective.

The optimal model was Adam & GA, trained on the novel dataset, which demonstrated remarkable performance in the simulation. The model demonstrated the capacity to complete the entire circuit with a top speed of 18.9 m/s, signifying a notable enhancement in performance compared to previous trials. In comparison, Adam also completed the stages but deviated from the road on the Jungle Track and then recovered and continued.

Table 4. Performance metrics for simulation on Jungle and Lake Tracks along 3 laps: Comparison of performance of the different CNN controllers, both original ones with standard optimizers and those improved applying GA. The results show the following: the used optimizer, the epoch (G is for considered generations in the genetic algorithm), and the obtained loss in the steering angle prediction. First line refers to original dataset and second to the new dataset. Together with the loss, the controller's performance in each track is presented as follows: driving speed (maximum, average, and standard deviation), distance covered out of the 3 laps, and the time driving (in seconds) to cover that distance.

Track	Optimizer	Epochs	Loss	Speed (m/s)			Dist. (m)	Dur. (s)
				Max	Avg	SD		
Lake (Training Track)	Adam & GA	50 G	0.0752	6.19	5.44	± 0.04	3 Laps	1400
		50 G	0.1199	7.92	7.48	± 0.059	3 Laps	1024
	Adam	10 E	0.0839	5.48	5.44	± 0.008	3 Laps	1400
		50 E	0.1229	8.03	7.48	± 0.060	3 Laps	1025
	Nadam & GA	50 G	0.0747	8.56	7.49	± 3.17	3 Laps	1032
		50 G	0.127	7.64	7.48	± 0.039	3 Laps	1021
	Nadam	10 E	0.0802	5.71	5.44	± 0.020	3 Laps	1400
		34 E	0.1298	7.64	7.48	± 0.027	3 Laps	1019
	RMSprop & GA	50 G	0.0709	6.31	5.41	± 0.35	863.122	357
		50 G	0.122	7.87	7.48	± 0.05	3 Laps	1023
	RMSprop	10 E	0.0755	7.68	5.42	± 0.40	737.042	304
		38 E	0.1291	7.79	7.48	± 0.05	3 Laps	1024
	Adadelata	10 E	0.1333	6.85	5.39	± 0.55	260.088	108
		50 E	0.2197	11.35	7.5	± 1.13	217.88	65
	Adamax	10 E	0.0889	6.32	5.37	± 0.49	384.058	160
		40 E	0.1339	7.68	7.48	± 0.04	3 Laps	1024
	SGD	10 E	0.1076	7.54	5.39	± 0.57	349.073	145
		31 E	0.1712	7.82	7.49	± 0.024	3 Laps	1016
	Adagrad	10 E	0.1084	6.76	5.37	± 0.62	218.067	91
		24 E	0.1908	11.77	7.51	± 0.89	339.08	101
	Ftrl	10 E	0.1563	7.08	5.35	± 0.58	265.392	111
		14 E	0.3008	9.94	7.4	± 0.81	254.91	77

Table 4. Cont.

Track	Optimizer	Epochs	Loss	Speed (m/s)			Dist. (m)	Dur. (s)
				Max	Avg	SD		
Jungle (Testing Track)	Adam & GA	50 G	0.0752	7.19	4.87	± 0.64	308.864	142
		50 G	0.1199	18.9	6.9	± 1.31	3 Laps	635
	Adam	10 E	0.0839	7.16	4.88	± 0.94	117.789	54
		50 E	0.1229	18.49	5.45	± 2.94	3 Laps	814
	Nadam & GA	50 G	0.0747	10.67	7.11	± 0.9	330.408	104
		50 G	0.127	9.66	6.89	± 1.57	110.844	36
	Nadam	10 E	0.0802	7.09	4.72	± 1.05	80.162	38
		34 E	0.1298	9.1651	6.63	± 2.12	26.694	9
	RMSprop & GA	50 G	0.0709	6.67	4.9	± 0.66	249.661	114
		50 G	0.122	18.21	7.3	± 1.86	777.966	238
	RMSprop	10 E	0.0755	7.12	4.89	± 0.67	238.48	109
		38 E	0.1291	10.3	6.98	± 1.18	287.352	92
	Adadelta	10 E	0.1333	9.38	4.89	± 2.94	21.85	10
		50 E	0.2197	10.08	6.74	± 2.37	7.528	9
	Adamax	10 E	0.0889	6.86	4.68	± 1.02	79.479	38
		40 E	0.134	10.15	7.12	± 0.61	141.3	160
	SGD	10 E	0.1076	10.82	4.98	± 1.43	80.204	36
		31 E	0.1712	9.896	7.12	± 0.88	216.49	68
	Adagrad	10 E	0.1084	7.25	4.69	± 1.19	62.94	30
		24 E	0.1908	9.5	7.1	± 0.96	171.264	54
	Ftrl	10 E	0.1563	7.01	4.45	± 1.89	19.916	10
		14 E	0.3008	9.93	6.69	± 2.6	20.937	7

5. Conclusions and Future Work

In this work, we present an approach for a module to be used in a future autonomous driving agent, aiming to infer the steering angle in a self-driving car based on a set of images captured while driving and received as input.

To this end, we created two of our own image-based datasets collected while driving in the Udacity simulator. The simulator allowed us to gathering images from three on-board camera locations (right, center, and left). Then, each group of images were preprocessed (to enhance their utility) and labeled considering the current steering angle.

Convolutional neural network (CNN) models of NVIDIA architecture with eight different optimizers (Nadam, Adam, RMSprop, Adamax, SGD, Adagrad, Adadelta, Ftrl) were trained on these data to analyze the best approach. Training the CNN model using the Nadam optimizer had the lowest/best average loss, while the highest/worst average loss was obtained using the Ftrl optimizer.

In addition, a genetic algorithm (GA) approach was proposed and tested, aiming to optimize the set of weights on which the CNN depends. Thus, additional CNN models based on the same architecture were then built starting from random weights. The GA was applied to optimize the weights of these models, as well as other previously trained and optimized models, using some of the aforementioned optimizers from the literature.

Several experiments were performed with different configurations for the GA, and the obtained models were then compared with previous optimizers. According to the results of these experiments, we concluded that the GA is a valid and promising tool for optimizing CNN weights, starting from both random values and optimized ones.

This study presents simulation results on the performance of CNN models using different optimization tools, such as Nadam, Adam, and Adam & GA (Adam with genetic algorithms). The results indicate that using Adam & GA was the most effective, while models using RMSprop were less effective than those using the genetic algorithm alone. The effectiveness of models using SGD, Adagrad, and Adadelta was found

to be weak. Furthermore, it was observed that models using Ftrl are not suitable for autonomous driving.

Thus, this tool will be deeply studied in future works, aiming to optimize not only weights of the models but also their internal structure (layer composition) and their configuration parameters (algorithm metaoptimization). In this line, the GA will be implemented using other tools and integrated into Keras, if possible, to take advantage of the computing resources that can be used in this framework. In addition, more sophisticated EA approaches (such as OpenAI-ES [35]) or even specialized ones will be applied to the CNN models, aiming to reach a better performance in autonomous driving. Another line to explore will be the consideration of Transformers [39], a more advanced deep learning technique that could be useful in this domain.

Author Contributions: Conceptualization, B.K. and A.M.M.; Methodology, B.K.; Software, B.K.; Validation, B.K. and A.M.M.; Formal Analysis, B.K.; Investigation, B.K.; Resources, B.K.; Data Curation, B.K.; Writing—Original Draft Preparation, B.K.; Writing—Review and Editing, A.M.M. and H.F.; Visualization, B.K.; Supervision, A.M.M. and H.F.; Project Administration, A.M.M.; Funding Acquisition, A.M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially funded by the Spanish Ministry of Science, Innovation and Universities MICIU/AEI/10.13039/501100011033 under project PID2023-147409NB-C21, as well as by the European Union NextGenerationEU/PRTR, under projects TED2021-131699B-I00 and TED2021-129938B-I00. It has also been funded by projects PID2020-113462RB-I00 and PID2020-115570GB-C22 of the Spanish Ministry of Economy and Competitiveness; project C-ING-179-UGR23 financed by the “Consejería de Universidades, Investigación e Innovación” (Andalusian Government, FEDER Program 2021-2027); and project PPJIA2023-031 (Plan Propio de Investigación y Transferencia UGR).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the reported results can be obtained upon reasonable request from the corresponding author.

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

References

1. Hagos, D.H.; Rawat, D.B. Recent Advances in Artificial Intelligence and Tactical Autonomy: Current Status, Challenges, and Perspectives. *Sensors* **2022**, *22*, 9916. [CrossRef] [PubMed]
2. Smolyakov, M.V.; Frolov, A.I.; Volkov, V.N.; Stelmashchuk, I.V. Self-Driving Car Steering Angle Prediction Based on Deep Neural Network: An Example of CarND Udacity Simulator. In Proceedings of the 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 17–19 October 2018; pp. 1–5. [CrossRef]
3. Khanum, A.; Lee, C.-Y.; Yang, C.-S. End-to-End Deep Learning Model for Steering Angle Control of Autonomous Vehicles. In Proceedings of the 2020 International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 13–16 November 2020; pp. 189–192. [CrossRef]
4. Valiente, R.; Zaman, M.; Ozer, S.; Fallah, Y.P. Controlling Steering Angle for Cooperative Self-Driving Vehicles Utilizing CNN and LSTM-Based Deep Networks. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2423–2428. [CrossRef]
5. Udacity. An Open Source Self-Driving Car. 2017. Available online: <https://udacity.com/self-driving-car> (accessed on 28 September 2021).
6. Lade, S.; Shrivastav, P.; Waghmare, S.; Hon, S.; Waghmode, S.; Teli, S. Simulation of Self-Driving Car Using Deep Learning. In Proceedings of the 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 5–7 March 2021; pp. 175–180. [CrossRef]
7. Setta, I.G.A.; Shehata, O.M.; Awad, M.A. Multivariate Prediction of Correct Lane for Autonomous Electric Vehicle Using Deep Learning Models. In Proceedings of the 2020 8th International Conference on Control, Mechatronics and Automation (ICCMA), Cairo, Egypt, 6–8 November 2020; pp. 127–130. [CrossRef]
8. Broome, M.; Gadd, M.; De Martini, D.; Newman, P. On the Road: Route Proposal from Radar Self-Supervised by Fuzzy LiDAR Traversability. *AI* **2020**, *1*, 558–585. [CrossRef]
9. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6999–7019. [CrossRef] [PubMed]

10. Coello, C.A.C. An Introduction to Evolutionary Algorithms and Their Applications. In *Advanced Distributed Systems*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3563, pp. 425–442.
11. Bäck, T. *Evolutionary Algorithms in Theory and Practice*; Oxford University Press: Oxford, UK, 1996.
12. Vikhar, P.A. Evolutionary Algorithms: A Critical Review and Its Future Prospects. In Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, 22–24 December 2016; pp. 261–265.
13. Mora, A.M.; Fernández-Ares, A.; Merelo, J.J.; García-Sánchez, P.; Fernndes, C.M. Effect of Noisy Fitness in Real-Time Strategy Games Player Behaviour Optimisation Using Evolutionary Algorithms. *J. Comput. Sci. Technol.* **2012**, *27*, 1007–1023. [CrossRef]
14. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison Wesley: Reading, MA, USA, 1989.
15. Mirjalili, S.; Song Dong, J.; Sadiq, A.S.; Faris, H. Genetic Algorithm: Theory, Literature Review, and Application in Image Reconstruction. In *Nature-Inspired Optimizers*; Mirjalili, S., Song Dong, J., Lewis, A., Eds.; Springer: Cham, Switzerland, 2020; Volume 811; p. 5_5. [CrossRef]
16. Tampuu, A.; Matiisen, T.; Semikin, M.; Fishman, D.; Muhammad, N. A Survey of End-to-End Driving: Architectures and Training Methods. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1364–1384. [CrossRef] [PubMed]
17. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, 24–28 September 2017; pp. 23–30.
18. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.
19. Malik, S.; Khan, M.A.; El-Sayed, H. CARLA: Car Learning to Act—An Inside Out. *Procedia Comput. Sci.* **2022**, *198*, 742–749. [CrossRef]
20. Amini, A.; Gilitschenski, I.; Phillips, J.; Moseyko, J.; Banerjee, R.; Karaman, S.; Rus, D. Learning Robust Control Policies for End-to-End Autonomous Driving from Data-Driven Simulation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1. [CrossRef]
21. Bewley, A.; Rigley, J.; Liu, Y.; Hawke, J.; Shen, R.; Lam, V.D.; Kendall, A. Learning to Drive from Simulation without Real World Labels. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20–24 May 2019; Volume 456, pp. 4818–4824.
22. Bojarski, M. End to End Learning for Self-Driving Cars. *arXiv* **2016**, arXiv:1604.07316.
23. Chishti, S.O.A.; Riaz, S.; Zaib, M.B.; Nauman, M. Self-Driving Cars Using CNN and Q-Learning. In Proceedings of the 2018 IEEE 21st International Multi-Topic Conference (INMIC), Karachi, Pakistan, 1–2 November 2018; pp. 1–7. [CrossRef]
24. Saranya, M.; Archana, N.; Reshma, J.; Sangeetha, S.; Varalakshmi, M. Object Detection and Lane Changing for Self-Driving Car Using CNN. In Proceedings of the 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 10–11 March 2022; pp. 1–7. [CrossRef]
25. Murillo-Morera, J.; Castro-Herrera, C.; Arroyo, J.; Fuentes-Fernandez, R. An Automated Defect Prediction Framework Using Genetic Algorithms: A Validation of Empirical Studies. *Intel. Artif.* **2016**, *19*, 114–137. Available online: <https://journal.iberamia.org/index.php/intartif/article/view/48> (accessed on 6 August 2024). [CrossRef]
26. Shafiee, M.J.; Chywl, B.; Li, F. Fast YOLO: A Fast You Only Look Once System for Real-Time Embedded Object Detection in Video. *arXiv* **2017**, arXiv:1709.00724. [CrossRef]
27. Buda, M.; Maki, A.; Mazurowski, M.A. A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks. *Neural Netw.* **2018**, *106*, 249–259. [CrossRef] [PubMed]
28. Fawole, O.A.; Rawat, D.B. Recent Advances in 3D Object Detection for Self-Driving Vehicles: A Survey. *AI* **2024**, *5*, 1255–1285. [CrossRef]
29. Lu, J.; Tan, L.; Jiang, H. Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. *Agriculture* **2021**, *11*, 707. [CrossRef]
30. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv* **2015**, arXiv:1511.07289.
31. Pagliuca, P.; Inglese, D.; Vitanza, A. Measuring emergent behaviors in a mixed competitive-cooperative environment. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2023**, *15*, 69–86.
32. Whitley, D.; Kauth, J. GENITOR: A different genetic algorithm. In Proceedings of the 4th Rocky Mountain Conference on Artificial Intelligence, Denver, CO, USA, 13–15 June 1988.
33. Hansen, N.; Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **2001**, *9*, 159–195. [CrossRef] [PubMed]
34. Wierstra, D.; Schaul, T.; Glasmachers, T.; Sun, Y.; Peters, J.; Schmidhuber, J. Natural evolution strategies. *J. Mach. Learn. Res.* **2014**, *15*, 949–980.
35. Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv* **2017**, arXiv:1703.03864.
36. Mean Squared Error. Wikipedia. Available online: https://en.wikipedia.org/wiki/Mean_squared_error (accessed on 22 October 2023).
37. Keras. Available online: <https://keras.io> (accessed on 5 June 2023).

38. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Available online: <https://tensorflow.org> (accessed on 5 June 2023).
39. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, Ł; Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*; 2017; Curran Associates, Inc.: Red Hook, NY, USA.

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.

Article

Efficient First-Order Algorithms for Large-Scale, Non-Smooth Maximum Entropy Models with Application to Wildfire Science

Gabriel Provencher Langlois ^{1,*}, Jatan Buch ² and Jérôme Darbon ³

¹ Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

² Department of Earth and Environmental Engineering, Columbia University, New York, NY 10027, USA; jb4625@columbia.edu

³ Division of Applied Mathematics, Brown University, Providence, RI 02912, USA; jerome_darbon@brown.edu

* Correspondence: gabriel.provencher@courant.edu

Abstract: Maximum entropy (MaxEnt) models are a class of statistical models that use the maximum entropy principle to estimate probability distributions from data. Due to the size of modern data sets, MaxEnt models need efficient optimization algorithms to scale well for big data applications. State-of-the-art algorithms for MaxEnt models, however, were not originally designed to handle big data sets; these algorithms either rely on technical devices that may yield unreliable numerical results, scale poorly, or require smoothness assumptions that many practical MaxEnt models lack. In this paper, we present novel optimization algorithms that overcome the shortcomings of state-of-the-art algorithms for training large-scale, non-smooth MaxEnt models. Our proposed first-order algorithms leverage the Kullback–Leibler divergence to train large-scale and non-smooth MaxEnt models efficiently. For MaxEnt models with discrete probability distribution of n elements built from samples, each containing m features, the stepsize parameter estimation and iterations in our algorithms scale on the order of $O(mn)$ operations and can be trivially parallelized. Moreover, the strong ℓ_1 convexity of the Kullback–Leibler divergence allows for larger stepsize parameters, thereby speeding up the convergence rate of our algorithms. To illustrate the efficiency of our novel algorithms, we consider the problem of estimating probabilities of fire occurrences as a function of ecological features in the Western US MTBS-Interagency wildfire data set. Our numerical results show that our algorithms outperform the state of the art by one order of magnitude and yield results that agree with physical models of wildfire occurrence and previous statistical analyses of wildfire drivers.

Keywords: primal–dual method; maximum entropy estimation; Kullback–Leibler divergence; wildfire science

MSC: 90C30; 90C06; 90C90; 62P12

1. Introduction

Maximum entropy (MaxEnt) models are a class of density estimation methods that use the maximum entropy principle [1] to reproduce key statistics of data sets. Historically used in physics [1,2], engineering [3–5], and statistics [6] applications, MaxEnt models are now frequently used for large-scale machine learning problems in natural language processing [7–12], social science [13–15], neuroscience [16–18], ecological modeling [19–28], and wildfire science [29–31]. MaxEnt models in machine learning must often estimate probability distributions from big data sets comprising hundreds of thousands to billions of samples or features or both [32]. Due to this enormous number, large-scale MaxEnt models need efficient and robust algorithms to perform well.

State-of-the-art algorithms for MaxEnt models, however, were not originally designed to handle big data sets; these algorithms either rely on technical devices that may yield unreliable numerical results [33], scale poorly in size [9,34], or require smoothness assumptions that many MaxEnt models lack in practice [10]. These limitations make it essentially

impossible to scale MaxEnt models to big data applications without adequate and costly computational resources [35,36]. This constraint on computational resources, in particular, has been recently identified as a crucial challenge to overcome for using large-scale MaxEnt models in climate change and wildfire studies within a reasonable amount of run time [27,28,30,37].

1.1. Contributions of This Paper

In this paper, we present novel optimization algorithms that overcome the shortcomings of state-of-the-art algorithms used for training large-scale, non-smooth MaxEnt models. Our proposed algorithms are first-order accelerated nonlinear primal–dual hybrid gradient (NPDHG) algorithms, whose theory was provided by two authors of this paper in [38], based on the Kullback–Leibler divergence. Using the Kullback–Leibler divergence over the classical proximal operator makes it possible to train large-scale and non-smooth MaxEnt models much more efficiently than the state of the art. In particular, we show that, for a MaxEnt model with a discrete probability distribution of n elements built from samples each containing m features, the stepsize parameter estimation and iterations in our algorithms all scale on the order of $O(mn)$ operations. This significantly improves on the known complexity bound of $O(\min(m^2n, mn^2))$ operations for computing the optimal stepsize parameters of classical first-order optimization methods, such as the linear PDHG or forward–backward splitting methods. We also show, as a consequence, that, for a given tolerance level $\epsilon > 0$, our algorithms provably compute solutions using on the order of $O(mn/\sqrt{\epsilon})$ or $O(mn/\log(1/\epsilon))$ operations, the order depending on the smoothness of the MaxEnt model and which are optimal with respect to the Nesterov class of optimal first-order methods [39]. Moreover, the computational bottleneck consists of matrix–vector multiplications, which can be trivially parallelized, and so our proposed algorithms exhibit scalable parallelism. Finally, we show that the strong convexity of the Kullback–Leibler divergence with respect to the ℓ_1 norm naturally allows for significantly larger stepsize parameters, thereby substantially speeding up the convergence rate of our algorithms.

To illustrate the efficiency of our novel algorithms on large-scale problems, we present an application to wildfire science. Specifically, we consider the problem of estimating the probabilities of fire occurrences across the western US as a function of ecological features. To do so, we fit elastic net, non-overlapping group lasso, and ℓ_∞ MaxEnt models to a large number of hyperparameters using our proposed algorithms and the state-of-the-art forward–backward splitting and coordinate descent STRUCTMAXENT2 algorithms [40,41]. Our numerical results show that our algorithms outperform the two latter algorithms by at least one order of magnitude, and yield results that are in good agreement with physical models of wildfire occurrence [42,43] as well as previous statistical analyses of wildfire drivers [29–31,44].

1.2. Organization of This Paper

This paper is organized as follows. In Section 2, we present our setup, we describe how MaxEnt models work, and we review three popular non-smooth MaxEnt models that are challenging to train with big data sets: the elastic net, group lasso, and ℓ_∞ regularized MaxEnt models. In Section 3, we explain why large-scale, non-smooth MaxEnt models are particularly challenging to train from big data sets, and we describe the state-of-the-art algorithms for training these MaxEnt models and their limitations. In Section 4, we describe our approach for estimating MaxEnt models with NPDHG optimization methods, derive explicit algorithms for regularized MaxEnt models, and explain why our algorithms overcome the limitations of state-of-the-art algorithms. In Section 5, we present an application of our algorithms to wildfire science, where we train a large number of MaxEnt models on the Western US MTBS-Interagency (WUMI) wildfire data set [45] to estimate the probabilities of fire occurrences as a function of ecological features. Finally, we review our results and outline directions for future work in Section 6.

2. Preliminaries

2.1. Setup

Suppose we receive l independent, identically distributed samples $\{v_1, \dots, v_l\} \subset \mathcal{I}$ from an unknown distribution $\mathcal{D}$. We assume throughout, without loss of generality, that the input space $\mathcal{I} \equiv \{1, \dots, n\}$. In addition, suppose we have a prior probability distribution p_{prior} on $\mathcal{I}$ that encapsulates some prior knowledge about the samples or unknown distribution. Finally, suppose we have access to a set of features from the samples via a bounded feature map $\Phi: \mathcal{I} \rightarrow \mathbb{R}^m$. Then, how do we estimate the unknown distribution $\mathcal{D}$ from the prior distribution p_{prior} , the samples $\{v_1, \dots, v_l\}$, and the feature map Φ ?

The maximum entropy principle offers a systematic way to answer this question. It states that the distribution that best estimates the unknown distribution $\mathcal{D}$ is the one that remains as close as possible to the prior probability p_{prior} while matching the features $\{\Phi(v_1), \dots, \Phi(v_l)\}$ exactly or as closely as possible, in some suitable sense. Specifically, let Δ_n denote the n -dimensional probability simplex and suppose $p_{\text{prior}} \in \text{int } \Delta_n$ (i.e., $p_{\text{prior}}(j) > 0$ for every $j \in \{1, \dots, n\}$). We measure the closeness of a probability distribution $p \in \Delta_n$ to the prior probability p_{prior} using the Kullback–Leibler divergence $D_{\text{KL}}: \Delta_n \times \text{int } \Delta_n \rightarrow \mathbb{R}$:

$$D_{\text{KL}}(p \parallel p_{\text{prior}}) = \sum_{j=1}^n p(j) \log(p(j)/p_{\text{prior}}(j)). \quad (1)$$

Next, let $\hat{\mathcal{D}}$ denote the empirical distribution induced by samples $\{v_1, \dots, v_l\}$:

$$\hat{\mathcal{D}}(j) = \frac{1}{l} |\{1 \leq i \leq l \mid v_i = j\}|. \quad (2)$$

In addition, let $\mathbb{E}_{\hat{\mathcal{D}}}[\Phi]$ denote the empirical average of the features induced by the samples and let $\mathbb{E}_p[\Phi]$ denote the model average induced by a probability distribution $p \in \Delta_n$:

$$\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] = \sum_{j=1}^n \hat{\mathcal{D}}(j) \Phi(j) \quad \text{and} \quad \mathbb{E}_p[\Phi] = \sum_{j=1}^n p(j) \Phi(j).$$

We measure how the averages $\mathbb{E}_{\hat{\mathcal{D}}}[\Phi]$ and $\mathbb{E}_p[\Phi]$ match using a function $H^*: \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ called the potential function. MaxEnt models combine the two measures described above and seek to minimize the sum

$$\inf_{p \in \Delta_n} f(p; t) = \inf_{p \in \Delta_n} \left\{ D_{\text{KL}}(p \parallel p_{\text{prior}}) + t H^* \left(\frac{\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_p[\Phi]}{t} \right) \right\} \quad (3)$$

where $t > 0$ is a hyperparameter selected using data-driven methods, e.g., cross-validation.

Standing assumptions: The potential function is proper, lower semicontinuous, convex, and bounded from below by zero with $H^*(\mathbf{0}) = 0$. Under the standing assumptions above on the potential function, which are fairly general, the probability distribution $\hat{\mathcal{D}}$ is a feasible point of (3). Moreover, the MaxEnt estimation problem (3) admits a unique global solution (see ([46], Page 35, Proposition 1.2); uniqueness follows from the strong convexity of the Kullback–Leibler divergence with respect to the ℓ_1 norm [47–51]).

2.2. Dual Formulation and Optimality Conditions

The generalized MaxEnt problem (3) admits a dual problem corresponding to regularized maximum a posteriori estimation [52]. To derive the dual problem, we first write the second term on the right hand side of (3) in terms of its convex conjugate,

$$t H^* \left(\frac{\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_p[\Phi]}{t} \right) = \sup_{w \in \mathbb{R}^m} \{ \langle w, \mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_p[\Phi] \rangle - t H(w) \},$$

where we abuse notation and write the convex conjugate of H^* as H . This lets us express problem (3) in saddle-point form:

$$\inf_{p \in \Delta_n} \sup_{w \in \mathbb{R}^m} \{ \langle w, \mathbb{E}_{\mathcal{D}}[\Phi] - \mathbb{E}_p[\Phi] \rangle - tH(w) + D_{\text{KL}}(p \parallel p_{\text{prior}}) \}. \quad (4)$$

Due to our assumptions on the potential function H^* , we can swap the infimum and supremum ([46], Statement (4.1) on page 61). Using the convex conjugate formula

$$\inf_{p \in \Delta_n} \{ D_{\text{KL}}(p \parallel p_{\text{prior}}) - \langle w, \mathbb{E}_p[\Phi] \rangle \} = -\log \left[\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle w, \Phi(j) \rangle} \right],$$

we obtain the dual problem of (3)

$$\sup_{w \in \mathbb{R}^m} \left\{ \langle w, \mathbb{E}_{\mathcal{D}}[\Phi] \rangle - tH(w) - \log \left[\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle w, \Phi(j) \rangle} \right] \right\}. \quad (5)$$

The dual problem (5) is a regularized maximum likelihood estimation problem over the family of Gibbs distributions [41,52]. It has at least one global solution ([46], Proposition (1.2) on page 35). Moreover, if (p^s, w^s) denotes a pair of solutions to the MaxEnt estimation problem (3) and its dual (5), then this pair satisfies the optimality conditions

$$\mathbb{E}_{\mathcal{D}}[\Phi] - \mathbb{E}_{p^s}[\Phi] \in t\partial H(w^s) \quad \text{and} \quad p^s(j) = \frac{p_{\text{prior}}(j) e^{\langle w^s, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle w^s, \Phi(j) \rangle}}, \quad (6)$$

for every $j \in \{1, \dots, n\}$ where $\partial H(w^s)$ denotes the subdifferential of H at w^s .

2.3. Examples of Non-Smooth MaxEnt Models

MaxEnt models differ in the choice of the prior distribution p_{prior} , hyperparameter t , and the potential function H^* . In practice, the prior distribution is often chosen to be uniform and the hyperparameter is chosen via data-driven methods. The choice of potential function depends on the application. The classical MaxEnt model uses the indicator function

$$u \mapsto H^*(u) = \begin{cases} 0, & \text{if } u = \mathbf{0}, \\ +\infty & \text{otherwise.} \end{cases}$$

This forces the model average $\mathbb{E}_p[\Phi]$ to be equal to the empirical average $\mathbb{E}_{\mathcal{D}}[\Phi]$. However, this is often too restrictive because the averages are expected to be close, and not equal, with high probability. Forcing the model and empirical averages to be equal can lead to severe over-fitting issues with big data sets, and more flexibility is therefore desired.

We review below three different non-smooth MaxEnt models that offer this flexibility: the elastic net, group lasso, and ℓ_∞ -regularized MaxEnt models. We chose these models because they are used extensively in statistics and machine learning but are also challenging to train on big data sets due to their non-smoothness [20,26,27,53]. These models will also be used for our numerical experiments and results in Section 5.

2.3.1. Elastic Net Regularized MaxEnt Models

These models use for potential function the relaxation of the convex conjugate of the ℓ_1 -norm:

$$u \mapsto H^*(u) \equiv \left(\frac{1-\alpha}{2} \|\cdot\|_2^2 + \alpha \|\cdot\|_1 \right)^* (u) = \frac{1}{2(1-\alpha)} \sum_{i=1}^m \max(0, |u_i| - \alpha)^2$$

where $\alpha \in (0, 1)$. This yields the elastic net regularized maximum entropy estimation problem

$$\min_{p \in \Delta_n} \left\{ D_{\text{KL}}(p \parallel p_{\text{prior}}) + \frac{1}{2t(1-\alpha)} \sum_{i=1}^m \max(0, |[\mathbb{E}_{\mathcal{D}}[\Phi]]_i - [\mathbb{E}_p[\Phi]]_i| - t\alpha)^2 \right\}. \quad (7)$$

This type of regularization is frequently used for feature selection in machine learning [12,20,54,55]. The corresponding dual problem is

$$\sup_{\mathbf{w} \in \mathbb{R}^m} \left\{ \langle \mathbf{w}, \mathbb{E}_{\mathcal{D}}[\Phi] \rangle - t \left(\frac{(1-\alpha)}{2} \|\mathbf{w}\|_2^2 + \alpha \|\mathbf{w}\|_1 \right) - \log \left[\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle \mathbf{w}, \Phi(j) \rangle} \right] \right\}. \quad (8)$$

Thanks to the elastic net penalty $\frac{(1-\alpha)}{2} \|\mathbf{w}\|_2^2 + \alpha \|\mathbf{w}\|_1$, problem (8) is strongly concave and has a unique solution.

An important aspect of the elastic net penalty is that it promotes sparsity; that is, it leads to a solution with many entries equal to zero [56–58], the number depending on the hyperparameter t and parameter α . Sparsity is useful when seeking accurate solutions based only on a few features or when performing feature selection.

2.3.2. Group Lasso Regularized MaxEnt Models

These models are specified starting from the dual problem. Consider a partition G of the components of a vector $\mathbf{w} \in \mathbb{R}^m$ into g different and possibly overlapping groups $[\mathbf{w}_1, \dots, \mathbf{w}_g]$ with $\mathbf{w}_g \in \mathbb{R}^{m_g}$ and $\cup_{g=1}^G \mathbf{w}_g = \mathbf{w}$. Let $\mathbf{w} \mapsto \|\mathbf{w}_g\|_{2,g}$ denote the ℓ_2 norm over the respective components of the g th group. Then, the dual version of the group lasso regularized maximum entropy estimation problem is

$$\sup_{\mathbf{w} \in \mathbb{R}^m} \left\{ \langle \mathbf{w}, \mathbb{E}_{\mathcal{D}}[\Phi] \rangle - t \sum_{g=1}^G \sqrt{m_g} \|\mathbf{w}_g\|_{2,g} - \log \left[\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle \mathbf{w}, \Phi(j) \rangle} \right] \right\}. \quad (9)$$

Thanks to the group lasso penalty $\sum_{g=1}^G \sqrt{m_g} \|\mathbf{w}_g\|_{2,g}$, problem (10) is strictly concave and so has at least one global solution. The corresponding primal problem follows from the convex conjugate formula

$$\mathbf{u} \mapsto H^*(\mathbf{u}) \equiv \left(\sum_{g=1}^G \sqrt{m_g} \|\cdot\|_{2,g} \right)^* (\mathbf{u}) = \begin{cases} 0, & \text{if } \|\mathbf{u}_g\|_{2,g} \leq \sqrt{m_g} \text{ for } g = 1, \dots, G, \\ +\infty, & \text{otherwise,} \end{cases}$$

and it is therefore given by

$$\min_{p \in \Delta_n} D_{\text{KL}}(p \parallel p_{\text{prior}}) \quad \text{s.t.} \quad \|\mathbb{E}_{\mathcal{D}}[\Phi_g] - \mathbb{E}_p[\Phi_g]\|_{2,g} \leq t\sqrt{m_g} \quad \text{for } g = 1, \dots, G. \quad (10)$$

Similarly to the elastic net penalty, the group lasso penalty promotes sparsity in the global solutions of (9), but it differs in that it sets different blocks of components $[\mathbf{w}_1, \dots, \mathbf{w}_g]$ to zero at the same time.

2.3.3. ℓ_∞ -Regularized MaxEnt Models

These models use for potential function the convex conjugate of the ℓ_∞ norm,

$$\mathbf{u} \mapsto H^*(\mathbf{u}) \equiv \|\mathbf{u}\|_\infty^* = \{\hat{\mathbf{u}} \in \mathbb{R}^m \mid \|\mathbf{u} - \hat{\mathbf{u}}\|_1 \leq 1\},$$

which is the characteristic set of the ℓ_1 ball with unit radius. This yields the ℓ_∞ -regularized maximum entropy estimation problem

$$\min_{p \in \Delta_n} \left\{ D_{\text{KL}}(p \parallel p_{\text{prior}}) + t \left\| \frac{\mathbb{E}_{\mathcal{D}}[\Phi] - \mathbb{E}_p[\Phi]}{t} \right\|_\infty^* \right\}. \quad (11)$$

This type of regularization is used for certain machine learning problems, including matrix factor decomposition, and in robust statistics applications (see, e.g., [53,59]). The corresponding dual problem is

$$\sup_{w \in \mathbb{R}^m} \left\{ \langle w, \mathbb{E}_{\mathcal{D}}[\Phi] \rangle - t \|w\|_{\infty} - \log \left[\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle w, \Phi(j) \rangle} \right] \right\} \quad (12)$$

Thanks to the ℓ_{∞} norm, problem (12) is strictly concave and has at least one global solution.

3. Related Work

3.1. Large-Scale Sparse MaxEnt Models: Computational Challenges

Estimating a probability distribution from the MaxEnt model (3) can be computationally prohibitive for big data sets. To illustrate this point, suppose the MaxEnt model (3) with hyperparameter $t \geq 0$ has a global solution $p^s(t)$. Let $p^{\epsilon}(t) \in \Delta_n$ with $\epsilon > 0$ denote an ϵ -approximate solution to the global solution $p^s(t)$, that is, the objective function f in (3) satisfies

$$f(p^{\epsilon}(t); t) - f(p^s(t); t) < \epsilon.$$

If the potential function H^* is smooth (equivalently, its conjugate H is strongly convex), the best achievable rate of convergence for computing $p^{\epsilon}(t)$ with $t > 0$ in the Nesterov class of optimal first-order methods is linear $O(\log(1/\epsilon))$ in the number of operations [39]. If H is not smooth, then the optimal convergence rate is sublinear $O(1/\sqrt{\epsilon})$ in the number of operations.

These rates, while optimal, require carefully fine-tuned stepsize parameters. In classical first-order optimization algorithms, these stepsize parameters are fine-tuned using a precise estimate of the largest singular value of the linear operator $A: \Delta_n \rightarrow \mathbb{R}^m$ defined by

$$Ap = \sum_{j=1}^n p(j) \Phi(j) = \mathbb{E}_p[\Phi]. \quad (13)$$

Unfortunately, computing the largest singular value of the linear operator A in (13) accurately is often computationally expensive for large matrices due to its prohibitive cubic cost of $O(\min(m^2n, mn^2))$ operations [60]. In this situation, line search methods and other heuristics can sometime be employed to bypass this issue, but they typically slow down the convergence speed. Another approach is to compute a crude estimate of the largest singular value of the matrix A , but doing so significantly reduces convergence speed as well. In fact, even if the largest singular value can be computed quickly, the resulting stepsize parameters may be much smaller than what is permissible to maintain convergence, that is, the largest singular value itself may be a poor estimate for determining how large the stepsize parameters are allowed to be to maintain convergence. This point will be discussed in more detail in Section 4.1 when we describe our proposed NPDHG algorithms.

This issue makes solving the MaxEnt model (3) difficult and laborious. Even worse, in some applications, the appropriate value of the hyperparameter t in (3) is difficult to guess and must be selected by repeatedly solving (3) from a large pool of values of hyperparameters, a process that can become particularly time-consuming and resource intensive for big data sets. This issue has driven much research in developing robust and efficient algorithms to minimize computational costs and maximize model performance.

3.2. State-of-the-Art Methods for Large-Scale, Non-Smooth MaxEnt Models

State-of-the-art methods for computing solutions to large-scale, non-smooth MaxEnt models are based on coordinate descent algorithms [61–64] and first-order optimization algorithms such as forward–backward splitting [40,65,66]. We will discuss these methods below, but note that other types of methods have been developed for MaxEnt models; see, e.g., [10,41,67,68] for surveys and comparisons of different algorithms. In particular,

we will not consider second-order-based methods suitable for smooth MaxEnt models such as limited-memory BFGS algorithms [10,69] since this work focuses on non-smooth MaxEnt models.

Coordinate descent methods. The state of the art is a coordinate descent algorithm based on a technical device called infinitely weighted logistic regression (IWLR) [26,33]. The IWLR method approximates a MaxEnt model as a logistic regression model and then fits the approximate logistic regression model using an existing, efficient optimization algorithm for logistic regression. This indirect approach was proposed because efficient and scalable coordinate descent algorithms were already available for fitting logistic regression models, which in [33] was an earlier version of the GLMNET software package [63]. The IWLR method remains the state of the art for this reason. It is implemented, for example, in the MaxEnt package available in the R programming language [26,63].

The IWLR method, however, is an approximate technical device that is not guaranteed to work, a fact acknowledged by the authors who proposed the method [33], and therefore may not produce reliable numerical results. Coordinate descent algorithms themselves are generally non-parallelizable and often lack robustness and good convergence properties. For example, the aforementioned GLMNET software package approximates the logarithm term in logistic regression models with a quadratic term to fit the models efficiently. Without costly stepsize optimization, which GLMNET avoids to improve performance, the GLMNET implementation may not converge [62,70]. A case in point is that [71] provides two numerical experiments in which GLMNET does not converge.

Other coordinate descent algorithms have been developed to compute solutions to non-smooth MaxEnt models directly (see, e.g., [64,72]) but they are also generally non-parallelizable and often lack robustness and good convergence properties. Finally, another issue is that many coordinate descent algorithms, including GLMNET, were designed for sparse MaxEnt models (e.g., the elastic net and group lasso regularized MaxEnt models described in Section 2.3) and those algorithms depend on the sparsity of the model to converge quickly [54]. It would be desirable to have fast optimization methods for when the feature mapping (13) is dense, as this often occurs in practice.

First-order methods. First-order optimization algorithms such as the forward–backward splitting algorithm are popular because they are robust and can provably compute ϵ -approximate solutions of (3) with an optimal rate of convergence. However, as discussed in detail at the end of the previous subsection, achieving this rate of convergence requires fine-tuning the stepsize parameters using an accurate estimate of the largest singular value of the feature mapping (13), and this estimate is typically computationally expensive for large matrices. This problem makes classical first-order optimization algorithms inefficient and impractical in estimating probability densities from large-scale MaxEnt models.

Summary. The current state-of-the-art algorithms for estimating probability densities from MaxEnt models either produce unreliable numerical results, lack scalable parallelism, or scale poorly in size. These shortcomings in terms of robustness and efficiency make it challenging to use non-smooth, large-scale MaxEnt models in applications without access to adequate and costly computational resources. The next section presents novel, efficient, and robust accelerated NPDHG optimization methods that address these shortcomings.

4. Main Results

We describe here our approach for computing solutions to the generalized MaxEnt problem (3) using accelerated NPDHG optimization methods [38]. In addition to the standing assumptions on the potential function H^* , we assume we can compute efficiently the proximal operator of the convex conjugate of the potential function

$$\arg \min_{w \in \mathbb{R}^m} \left\{ \lambda H(w) + \frac{1}{2} \|w - \hat{w}\|_2^2 \right\},$$

for every $\lambda > 0$ and $\hat{w} \in \mathbb{R}^m$. These assumptions are satisfied for most potential functions used in practice, including for the potential functions described in Section 2.3.

4.1. Methodology

We start with the saddle-point formulation (4) of the generalized MaxEnt estimation problem (3). Based on the work two of the authors provided in [38], we propose to split the infimum and supremum in the saddle-point problem (4) using an iterative NPDHG scheme that alternates between a nonlinear proximal ascent step based on the Kullback–Leibler divergence and a linear proximal descent step.

More precisely, let $\tau_0 > 0$ and $\sigma_0 > 0$ be two stepsize parameters satisfying the inequality

$$\tau_0 \sigma_0 \left(\max_{j \in \{1, \dots, n\}} \|\Phi(j)\|_2^2 \right) \leq 1, \quad (14)$$

let $\theta_0 = 0$, let $w_{-1} = w_0 \in \text{int dom } H$, let $z_0 \in \mathbb{R}^m$, and define the initial probability distribution $p_0 \in \Delta_n$ through the constant z_0 via

$$p_0(j) = \frac{p_{\text{prior}}(j) e^{\langle z_0, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_0, \Phi(j) \rangle}} \quad (15)$$

for each $j \in \{1, \dots, n\}$. Our proposed NPDHG iterative scheme computes the iterates

$$\begin{aligned} p_{k+1} &= \arg \min_{p \in \Delta_n} \{ \tau_k D_{\text{KL}}(p \parallel p_{\text{prior}}) - \tau_k \langle w_k + \theta_k(w_k - w_{k-1}), \mathbb{E}_p[\Phi] \rangle + D_{\text{KL}}(p \parallel p_k) \} \\ \hat{w}_{k+1} &= w_k + \sigma_k (\mathbb{E}_{\hat{p}}[\Phi] - \mathbb{E}_{p_{k+1}}[\Phi]), \\ w_{k+1} &= \arg \min_{w \in \mathbb{R}^m} \left\{ t \sigma_k H(w) + \frac{1}{2} \|w - \hat{w}_{k+1}\|_2^2 \right\}, \\ \theta_{k+1} &= 1 / \sqrt{1 + \tau_k}, \quad \tau_{k+1} = \theta_{k+1} \tau_k \quad \text{and} \quad \sigma_{k+1} = \sigma_k / \theta_{k+1}. \end{aligned} \quad (16)$$

According to ([38], Proposition 4.1), the sequence of iterates p_k converges strongly to the unique solution of the generalized MaxEnt problem (3). Moreover, for any $t \geq 0$ and a given tolerance level $\epsilon > 0$, the scheme (16) provably computes an ϵ -approximate solution $p^\epsilon(t)$ of the generalized MaxEnt model (3) in $O(1/\sqrt{\epsilon})$ time. This rate of convergence is, without further smoothness assumptions on the potential function H^* , the best achievable rate of convergence with respect to the Nesterov class of optimal first-order methods [39].

The key element in this scheme is the choice of the Kullback–Leibler divergence as a nonlinear proximal step in the first line of (16). We use it for two reasons: First, because the Kullback–Leibler divergence already appears in the saddle-point problem (4). This allows us to compute p_{k+1} explicitly. Indeed, thanks to the choice of initial probability distribution (15), we have

$$p_{k+1}(j) = \frac{p_{\text{prior}}(j) e^{\langle z_{k+1}, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_{k+1}, \Phi(j) \rangle}} \quad \text{with} \quad z_{k+1} = \frac{1}{1 + \tau_k} (z_k + \tau_k (w_k + \theta_k (w_k - w_{k-1}))) \quad (17)$$

for each $j \in \{1, \dots, n\}$. See Appendix A for the derivation of (17).

Second, because the Kullback–Leibler divergence is 1-strongly convex with respect to the ℓ_1 norm, that is,

$$D_{\text{KL}}(p \parallel p_k) \geq \frac{1}{2} \|p - p_k\|_1^2.$$

This fact follows from a fundamental result in information theory known as Pinsker’s inequality [47–51]. In particular, this means that scheme (16) alternates between solving a strongly convex problem over the space $(\mathbb{R}^n, \|\cdot\|_1)$ and a concave problem over the space

$(\mathbb{R}^m, \|\cdot\|_2)$. The choice of these spaces is significant, for the induced operator norm $\|\cdot\|_{\text{op}}$ of the linear operator A defined in (13) becomes

$$\|A\|_{\text{op}} = \sup_{\|p\|_1=1} \|Ap\|_2 = \max_{j \in \{1, \dots, n\}} \|\Phi(j)\|_2.$$

This operator norm offers two crucial advantages: First, it can be computed in optimal $\Theta(mn)$ time, or better if the features $\{\Phi(j)\}_{j=1}^n$ have structure, e.g., if they are sparse. This means that the stepsize parameters τ_0 and σ_0 of the NPDHG scheme can be computed in (14), with equality, in optimal $\Theta(mn)$ time. This is important because, in classical first-order optimization methods, we typically require a precise estimate of the largest singular value of the feature mapping (13), namely the number

$$\|A\|_2 = \sup_{\|x\|_2=1} \|Ax\|_2,$$

to fine-tune the stepsize parameters to gain an optimal convergence rate. However, as discussed in detail in Section 3.1, this estimate is computationally expensive for large matrices due to its prohibitive cubic cost of $O(\min(m^2n, mn^2))$ operations. In contrast, our NPDHG scheme (16) does not suffer from this computational bottleneck and therefore scales well. Second, the operator norm $\|A\|_{\text{op}}$ can be significantly smaller than the estimate $\|A\|_2$, hence allowing for bigger stepsize parameters to further speed up convergence. (An easy calculation yields $\max_{\|p\|_2=1} \|Ap\|_2 \geq \max_{j \in \{1, \dots, n\}} \|\Phi(j)\|_2$.)

Summary. To solve the generalized MaxEnt estimation problem (3) and its dual problem (5), let $\tau_0 > 0$ and $\sigma_0 > 0$ be two stepsize parameters satisfying inequality (14), let $\theta_0 = 0$, let $w_{-1} = w_0 \in \text{int dom } H$, let $z_0 \in \mathbb{R}^m$, and define an initial probability distribution p_0 through z_0 via (15). Then, compute the iterates

$$\begin{aligned} z_{k+1} &= (z_k + \tau_k(w_k + \theta_k(w_k - w_{k-1}))) / (1 + \tau_k), \\ p_{k+1}(j) &= \frac{p_{\text{prior}}(j)e^{\langle z_{k+1}, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j)e^{\langle z_{k+1}, \Phi(j) \rangle}} \quad \text{for } j \in \{1, \dots, n\}, \\ \hat{w}_{k+1} &= w_k + \sigma_k(\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p_{k+1}}[\Phi]), \\ w_{k+1} &= \arg \min_{w \in \mathbb{R}^m} \left\{ t\sigma_k H(w) + \frac{1}{2} \|w - \hat{w}_{k+1}\|_2^2 \right\}, \\ \theta_{k+1} &= 1/\sqrt{1 + \tau_k}, \quad \tau_{k+1} = \theta_{k+1}\tau_k \quad \text{and} \quad \sigma_{k+1} = \sigma_k/\theta_{k+1}, \end{aligned} \tag{18}$$

until convergence is achieved. As all parameters and updates can be computed in $O(mn)$ time, for any $t \geq 0$ and a given tolerance level $\epsilon > 0$, the overall complexity for computing an ϵ -approximate solution $p^\epsilon(t)$ is $O(mn/\sqrt{\epsilon})$.

4.2. Algorithm for Smooth Potential Functions

The iterative scheme (18) does not require the potential function H^* to be smooth. If, however, the potential function H^* is γ_{H^*} -smooth (equivalently, H is $\frac{1}{\gamma_{H^*}}$ -strongly convex) for some $\gamma_{H^*} > 0$, then we can modify the NPDHG iterative scheme (18) to achieve a linear rate of convergence. More precisely, let $t > 0$ and introduce the stepsize parameters

$$\theta = 1 - \frac{t}{2\gamma_{H^*}\|A\|_{\text{op}}^2} \left(\sqrt{1 + \frac{4\gamma_{H^*}\|A\|_{\text{op}}^2}{t}} - 1 \right), \quad \tau = \frac{1 - \theta}{\theta} \quad \text{and} \quad \sigma = \frac{\gamma_{H^*}\tau}{t}. \tag{19}$$

Let $w_{-1} = w_0 \in \text{int dom } H$, let $z_0 \in \mathbb{R}^m$, and define p_0 through z_0 via (15). Then, the explicit NPDHG iterative scheme is

$$\begin{aligned} z_{k+1} &= (z_k + \tau(w_k + \theta(w_k - w_{k-1}))) / (1 + \tau), \\ p_{k+1}(j) &= \frac{p_{\text{prior}}(j)e^{\langle z_{k+1}, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j)e^{\langle z_{k+1}, \Phi(j) \rangle}} \quad \text{for } j \in \{1, \dots, n\}, \\ \hat{w}_{k+1} &= w_k + \sigma(\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p_{k+1}}[\Phi]), \\ w_{k+1} &= \arg \min_{w \in \mathbb{R}^m} \left\{ t\sigma H(w) + \frac{1}{2} \|w - \hat{w}_{k+1}\|_2^2 \right\}. \end{aligned} \tag{20}$$

According to ([38], Proposition 4.3), the sequences of iterates p_k and w_k converge strongly to the unique solution of the generalized MaxEnt estimation problem (3) and its dual problem (5). Moreover, for any $t > 0$ and a given tolerance level $\epsilon > 0$, this scheme provably computes an ϵ -approximate solution $p^\epsilon(t)$ of the generalized MaxEnt estimation problem (3) in $O(\log(1/\epsilon))$. This rate of convergence is the best achievable rate of convergence with respect to the Nesterov class of optimal first-order methods [39]. As all parameters and updates can be computed in $O(mn)$ time, the overall complexity for computing an ϵ -approximate solution $p^\epsilon(t)$ is $O(mn \log(1/\epsilon))$.

5. Application to Wildfire Science

To illustrate the efficiency of our novel algorithms on large-scale problems, we present here an application to wildfire science. The problem at hand is to combine fire occurrence data with ecological data in a fixed geographical region to estimate the probability of fire occurrences as a function of ecological features. MaxEnt models achieve this goal by translating fire occurrence and ecological data into probabilities of fire occurrences and ecological features. This approach closely mirrors how MaxEnt models are used for modeling species' geographic distributions [19,20,22,24–28]. Another related goal is to identify what ecological features correlate most with fire occurrences. This can be achieved using a sparse MaxEnt model, e.g., an elastic net or group lasso regularized MaxEnt model, to identify ecological features correlating significantly with fire occurrences.

For this application, we use the Western US MTBS-Interagency (WUMI) wildfire data set [45], which we describe in Section 5.1 below. We formulate the problem of combining the fire occurrence and ecological data from the WUMI wildfire data set into a MaxEnt estimation problem in Section 5.2. Using this data, we then fit the elastic net, (non-overlapping) group lasso, and ℓ_∞ MaxEnt models for a large number of hyperparameters weighting the regularization. We detail this fitting procedure and the explicit NPDHG algorithms for these MaxEnt models in Section 5.3. Following this, we compare in Section 5.4 the running times required to fit the aforementioned MaxEnt models using our NPDHG algorithms with the forward-backward splitting algorithm [40,65] and the STRUCTMAXENT2 coordinate descent algorithm of [41]. Finally, in Section 5.5, we interpret the results obtained from fitting the aforementioned MaxEnt models to the WUMI wildfire data set.

5.1. WUMI Wildfire Data Set

The Western US MTBS-Interagency wildfire data set [45] consists of all fires ($\geq 1 \text{ km}^2$) from multiple state and federal agencies, supplemented by satellite observations of large fires ($\geq 4 \text{ km}^2$) from the Monitoring Trends in Burn Severity (MTBS) program, in the continental United States west of 103° W longitude. For this application, we extracted all wildfires from the WUMI data set that occurred between 1984 and 2020 inclusive (accessed 18 May 2023). The locations of all fires used are shown in Figure 1.

Next, following the procedure outlined in [44], we overlay the fire locations on a $12 \text{ km} \times 12 \text{ km}$ grid to construct a data frame of prevailing climate, vegetation, topographic, and human-related features for each fire. Altogether we include a total of 35 potential

fire-related features from various sources; we provide a summary of all features used in our analysis along with their sources in Appendix B. For each grid cell, we also provide an index identifying its Environmental Protection Agency (EPA) Level III ecoregion. Defined on the basis of roughly homogeneous climate and vegetation patterns, ecoregions are commonly used in the wildfire science literature to identify climate–fire relationships at a coarse spatial scale [29,73]. The full WUMI wildfire data set with associated fire-related features for all months from 1984 to 2020 is publicly available as part of the Stochastic Machine Learning for wildfire activity in the western US (SMLFire1.0) package [44].

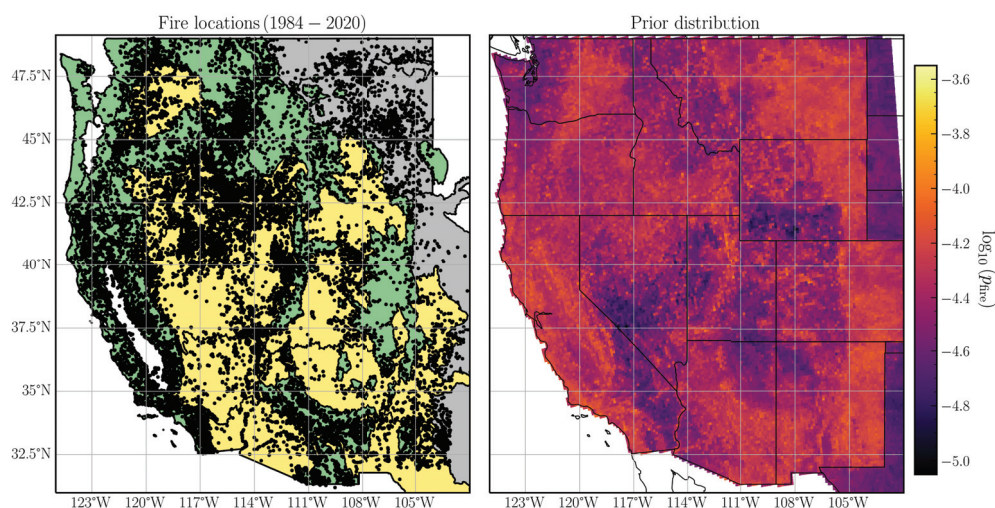


Figure 1. Wildfire activity in the western United States from 1984 to 2020. **(Left)** Fire locations of all fires (black dots) in the Western US MTBS-Interagency (WUMI) data set; also shown are three ecological divisions characterized by their primary vegetation type—forests (green), deserts (yellow), and plains (gray). **(Right)** Prior distribution indicating mean fire probability across all calendar months.

5.2. Data Preprocessing

We construct a weakly informative prior [74,75] for incorporating existing knowledge of wildfire conducive environmental conditions in our MaxEnt model. First, we prepare a training data set of 10,000 fire occurrences and absences chosen randomly across all months between 1984 and 2020 correlated with the values of 35 fire-related features described in the previous section. The features are averaged over each calendar month (i.e., January, February, ...) between 1984 and 2020. We then apply min–max scaling to each feature, ensuring that all features lie in the same range. Second, we construct two Random Forest (RF) models, one each for fires in dry (May–September) and wet (October–April) seasons respectively. We withhold 20% of the training data to tune the model hyperparameters, such that the optimal hyperparameters ensure a trade-off between model precision and recall. Next, we predict the fire probability for all grid cells with fire-related features using either the trained wet or dry season RF model depending on the calendar month. Since the MaxEnt algorithm assumes a presence-only framework, that is, the absence of fires in a grid cell does not imply a non-zero fire probability, we impute grid cells without a fire with probability $p_{\text{nfire}} = \min(p_{\text{fire}})/10$. Last, we normalize the prior probability distribution to ensure that the fire probability across all grid cells sums up to one. For convenience, we represent the prior distribution as the normalized mean of 12 monthly fire probabilities predicted by the RF models in Figure 1.

To construct the empirical distribution $\hat{D}$, we divide the study region into its 18 different EPA level III ecoregions and weigh the relative frequencies of fire among the ecoregions using the following strategy. For each ecoregion $r \in \{1, \dots, 18\}$, let $n_{r,\text{total}}$ denote the total number of grid cells in ecoregion r and let $n_{r,\text{fire}}$ denote the total number grid cell in ecoregion r where at least one fire was recorded. In addition, let $Z = \sum_{r=1}^{18} \frac{n_{r,\text{fire}}}{n_{r,\text{total}}}$ denote the sum

of the proportions of grid cells where at least one fire was recorded among all ecoregions. Then, we compute the empirical probability at cell $j \in \{1, \dots, n\}$, $\hat{D}(j)$, as follows:

$$\hat{D}(j) = \frac{1}{Z} \begin{cases} 0 & \text{if no fire was recorded in grid cell } j, \\ 1/n_{r,\text{total}} & \text{if at least one fire was recorded in grid } j \text{ and the grid} \\ & \text{cell } j \text{ belongs to ecoregion } r \end{cases} \quad (21)$$

This construction gives the empirical distribution more weight to ecoregions where fires are more frequent and widespread.

5.3. Fitting Procedure and Algorithmic Setup

For the analysis, we fit six different MaxEnt models to the wildfire data: four elastic net MaxEnt models with parameters $\alpha = \{0.95, 0.4, 0.15, 0.05\}$, a non-overlapping group lasso MaxEnt model, and an ℓ_∞ -regularized MaxEnt model. For each model, we fit a regularization path of 141 hyperparameters as follows:

$$t^{(l)} = \begin{cases} (1 - l/100)t^{(0)} & \text{from } l = 0 \text{ to } 50 \\ (0.5 - (l - 50)/200)t^{(0)} & \text{from } l = 51 \text{ to } 140, \end{cases}$$

where $t^{(0)}$ depends on the MaxEnt model and corresponds to the smallest hyperparameter for which the primal and dual solutions are equal to the prior distribution and zero. More precisely, for each model we set $(p^{(0)}, w^{(0)}) = (p_{\text{prior}}, \mathbf{0})$ and compute the sequence of solutions $\{(p^{(l)}, w^{(l)})\}_{l=1}^{140}$ of the corresponding MaxEnt primal and dual problems (7) and (8) with hyperparameter $t = t^{(l)}$.

We chose these MaxEnt models to study the impact of the regularization on the primal and dual solutions as a function of the hyperparameters. In particular, for the elastic net and non-overlapping group lasso MaxEnt models, we are interested in identifying the set of features that are selected or discarded as a function of the sequence of hyperparameters $\{t^{(l)}\}_{l=0}^{140}$. In the parts below, we describe the value of $t^{(0)}$ and the NPDHG algorithm used for each model.

5.3.1. Elastic Net MaxEnt Models

For these MaxEnt models, the smallest hyperparameter for which the solutions to the primal and dual problems (7) and (8) are equal to the prior distribution and zero is

$$t^{(0)} = \left\| \mathbb{E}_{\hat{D}}[\Phi] - \mathbb{E}_{p_{\text{prior}}}[\Phi] \right\|_\infty / \alpha.$$

This follows from the optimality condition

$$\mathbb{E}_{\hat{D}}[\Phi] - \mathbb{E}_{p^s}[\Phi] - t(1 - \alpha)w^s \in t\alpha\partial\|\cdot\|_1 w^s \quad (22)$$

and computing the smallest parameter t for which (22) is satisfied at $(p^s, w^s) = (p_{\text{prior}}, \mathbf{0})$.

For the elastic net MaxEnt model with $\alpha = 0.95$, we use the NPDHG algorithm (18) (with sublinear convergence rate), while, for the models with $\alpha = \{0.4, 0.15, 0.05\}$, we use the NPDHG algorithm (20) (with linear convergence rate). Starting from $l = 1$, we compute the pair of solutions $(p^{(l)}, w^{(l)})$ to the primal and dual problems (7) and (8) at hyperparameter $t = t^{(l)}$ using the previously computed pair of solutions $(p^{(l-1)}, w^{(l-1)})$ by setting the initial vectors $z_0 = w_0 = w^{(l-1)}$ in both NPDHG algorithms. For the stepsize parameters, we set $\theta_0 = 0$, $\tau_0 = 2$, and $\sigma_0 = 1/2\|A\|_{\text{op}}^2$ in (18), and we set θ , τ , and σ according to the formulas in (19) with $\gamma_{H^*} = 1 - \alpha$ in (20). We compute the update

w_{k+1} in both NPDHG algorithms using the classical soft thresholding operator [66,76,77]. Specifically, for any $\lambda > 0$, $\hat{w} \in \mathbb{R}^m$, and $i \in \{1, \dots, m\}$ we have

$$\left[\arg \min_{w \in \mathbb{R}^m} \left\{ \lambda \|w\|_1 + \frac{1}{2} \|w - \hat{w}\|_2^2 \right\} \right]_i \equiv [\text{shrink}_1(\hat{w}, \lambda)]_i = \begin{cases} [\hat{w}]_i - \lambda & \text{if } [\hat{w}]_i > \lambda, \\ 0 & \text{if } |[\hat{w}]_i| \leq \lambda, \\ [\hat{w}]_i + \lambda & \text{if } [\hat{w}]_i < -\lambda, \end{cases}$$

and so, for every $\lambda > 0$, $\alpha \in [0, 1]$, $\hat{w} \in \mathbb{R}^m$ and $i \in \{1, \dots, m\}$ we have

$$\left[\arg \min_{w \in \mathbb{R}^m} \left\{ \lambda \left(\alpha \|w\|_1 + \frac{(1-\alpha)}{2} \|w\|_2^2 \right) + \frac{1}{2} \|w - \hat{w}\|_2^2 \right\} \right]_i = \frac{[\text{shrink}_1(\hat{w}, \lambda \alpha)]_i}{1 + \lambda(1-\alpha)}.$$

Finally, we let the NPDHG algorithms run for at least 40 iterations before checking for convergence. We stop the NPDHG algorithms when the optimality condition (22) is satisfied within some tolerance 10^{-5} :

$$\|\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p_k}[\Phi] - t(1-\alpha)w_k\|_\infty \leq t\alpha(1+10^{-5}).$$

5.3.2. Non-Overlapping Group Lasso MaxEnt Model

For this MaxEnt model, we divide the features into five disjoint groups of features, as described in Appendix B. Then, the smallest hyperparameter for which the solutions to the primal and dual problems (7) and (8) are equal to the prior distribution and zero is

$$t^{(0)} = \max_{g \in \{1, \dots, 5\}} \left\{ \|\mathbb{E}_{\hat{\mathcal{D}}}[\Phi_g] - \mathbb{E}_p[\Phi_g]\|_{2,g} / \sqrt{m_g} \right\},$$

where m_g is the number of features in the g th group. This follows from the optimality condition

$$\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p^s}[\Phi] \in \bigcup_{g=1}^5 \left\{ t\sqrt{m_g} \partial \|\cdot\|_{2,g} \right\} w_g^s \quad (23)$$

and computing the smallest parameter t for which (23) is satisfied at $(p^s, w^s) = (p_{\text{prior}}, \mathbf{0})$.

For this model, we use the NPDHG algorithm (18). Starting from $l = 1$, we compute the pair of solutions $(p^{(l)}, w^{(l)})$ to the primal and dual problems (10) and (9) at hyperparameter $t = t^{(l)}$ using the previously computed pair of solutions $(p^{(l-1)}, w^{(l-1)})$ by setting the initial vectors $z_0 = w_0 = w^{(l-1)}$. For the stepsize parameters, we set $\theta_0 = 0$, $\tau_0 = 2$, and $\sigma_0 = 1/2 \|A\|_{\text{op}}^2$. We compute the update w_{k+1} in (18) using the following proximal operator formula: for every group $g \in \{1, \dots, 5\}$, $\lambda > 0$ and $\hat{w} \in \mathbb{R}^m$,

$$\arg \min_{w \in \mathbb{R}^m} \left\{ \lambda \sqrt{m_g} \|w\|_{2,g} + \frac{1}{2} \|w - \hat{w}\|_{2,g}^2 \right\} = \max(0, 1 - \lambda \sqrt{m_g} / \|\hat{w}\|_2) \hat{w}.$$

Finally, we let the NPDHG algorithms run for at least 40 iterations before checking for convergence. We stop the NPDHG algorithms when the optimality condition (23) is satisfied within some tolerance 10^{-5} :

$$\max_{g \in \{1, \dots, 5\}} \|\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p_k}[\Phi]\|_{2,g} \leq t(1+10^{-5}).$$

5.3.3. ℓ_∞ -Regularized MaxEnt Model

For this MaxEnt model, the smallest hyperparameter for which the solutions to the primal and dual problems (7) and (8) are equal to the prior distribution and zero is

$$t^{(0)} = \|\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p_{\text{prior}}}[\Phi]\|_1.$$

This follows from the optimality condition

$$\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p^s}[\Phi] \in t\partial\|\cdot\|_{\infty}(\mathbf{w}^s) \quad (24)$$

and computing the smallest parameter t for which (24) is satisfied at $(p^s, \mathbf{w}^s) = (p_{\text{prior}}, \mathbf{0})$.

For this model, we use the NPDHG algorithm (18). Starting from $l = 1$, we compute the pair of solutions $(p^{(l)}, \mathbf{w}^{(l)})$ to the primal and dual problems (11) and (12) at hyperparameter $t = t^{(l)}$ using the previously computed pair of solutions $(p^{(l-1)}, \mathbf{w}^{(l-1)})$ by setting the initial vectors $\mathbf{z}_0 = \mathbf{w}_0 = \mathbf{w}^{(l-1)}$. For the stepsize parameters, we set $\theta_0 = 0$, $\tau_0 = 2$, and $\sigma_0 = 1/2\|\mathbf{A}\|_{\text{op}}^2$. Finally, we compute the update $\mathbf{w}_{k+1}$ in (18) using Moreau's decomposition ([78], Theorem 3.2.5): for every $\lambda > 0$ and $\hat{\mathbf{w}} \in \mathbb{R}^m$,

$$\arg \min_{\mathbf{w} \in \mathbb{R}^m} \left\{ \lambda \|\mathbf{w}\|_{\infty} + \frac{1}{2} \|\mathbf{w} - \hat{\mathbf{w}}\|_2^2 \right\} = \hat{\mathbf{w}} - \arg \min_{\|\mathbf{w}\|_1 \leq \lambda} \frac{1}{2} \|\mathbf{w} - \hat{\mathbf{w}}\|_2^2.$$

The second term on the right amounts to projecting $\hat{\mathbf{w}}$ on the ℓ_1 ball of radius λ . There are fast algorithms for doing so; we use Algorithm 1 described in [79]. Finally, we let the NPDHG algorithms run for at least 40 iterations before checking for convergence. We stop the NPDHG algorithms when the optimality condition (24) is satisfied within some tolerance 10^{-5} :

$$\|\mathbb{E}_{\hat{\mathcal{D}}}[\Phi] - \mathbb{E}_{p_k}[\Phi]\|_1 \leq t(1 + 10^{-5}).$$

5.4. Comparison of Timings

In this section, we compare the run times of our NPDHG algorithms with two state-of-the-art optimization algorithms for solving non-smooth MaxEnt models: the forward-backward splitting algorithm (specifically, Algorithm 5 in [65]; see also [40,66]) and the STRUCTMAXENT2 coordinate descent algorithm from [41,64]. All numerical experiments were performed on a single core Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz.

We initially chose the GLMNET implementation [26,33] in MATLAB 2023b over STRUCTMAXENT2 for the numerical comparisons, but we found that GLMNET produced unreliable numerical solutions when compared to both the NPDHG and forward-backward splitting algorithms. We also tried using GLMNET's implementation in the R language, but to no avail. We think this problem arises because, as discussed in Section 3.2, the GLMNET algorithm approximates a MaxEnt model as a logistic regression model and then invokes a coordinate descent method tailored to logistic regression to approximate the solution to the MaxEnt model. Our observations suggest this approach does not work well for our data set. In contrast, we found that the STRUCTMAXENT2 algorithm produced correct numerical results.

For the forward-backward splitting algorithm, the stepsize parameters were set to $1/\|\mathbf{A}\|_2$ and 1 (corresponding to τ and t_0 in Algorithm 5 of [65]) and, for computing the pair of solutions $(p^{(l)}, \mathbf{w}^{(l)})$ at hyperparameter $t^{(l)}$, the initial iterate was set to $\mathbf{w}^{(l-1)}$. In addition, for the elastic net MaxEnt models, the acceleration quantity $q = (1 - \alpha)t^{(l)} / (\|\mathbf{A}\|_2 + (1 - \alpha)t^{(l)})$ was employed. We used the stopping criteria of the NPDHG algorithms for the forward-backward splitting algorithms. For the coordinate descent algorithm, we modified the STRUCTMAXENT2 algorithm from ([41], pp. 305–306) to make it applicable to the elastic net penalty. For computing the pair of solutions $(p^{(l)}, \mathbf{w}^{(l)})$ at hyperparameter $t^{(l)}$, the initial iterate was set to $\mathbf{w}^{(l-1)}$. We did not use the STRUCTMAXENT2 algorithm for the non-overlapping group lasso or ℓ_{∞} -regularized MaxEnt models, as it was not designed for these MaxEnt models. We used the stopping criteria of the NPDHG algorithms for the STRUCTMAXENT2 algorithm.

Table 1 shows the average timings for computing the entire regularization path of the WUMI wildfire data set using the coordinate descent, forward-backward splitting, and NPDHG algorithms. All timings were averaged over five runs and, for the forward-

backward splitting and NPDHG algorithms, they include the time required to compute all the stepsize parameters. All algorithms were implemented in MATLAB.

Table 1. Timings results (in seconds) for fitting the MaxEnt models described in Section 5.3. All timings are averaged over five runs.

	STRUCTMAXENT2	Forward–Backward Splitting	NPDHG
Elastic net ($\alpha = 0.95$)	5562.19	4208.01	365.55
Elastic net ($\alpha = 0.40$)	1018.73	1407.22	113.53
Non-overlapping group lasso	N/A	3036.38	278.14
ℓ_∞ -regularization	N/A	2534.65	289.98

The NPDHG algorithm outperformed both the forward–backward splitting and coordinate descent algorithms by at least one order of magnitude. In particular, we observed that the NPDHG algorithm required far fewer iterations to achieve convergence compared to both the forward–backward splitting and STRUCTMAXENT2 algorithms. This difference is because the stepsize parameters for the NPDHG algorithm were much larger compared to either the forward–backward splitting or the STRUCTMAXENT2 algorithm. Indeed, the stepsize parameters for the NPDHG and forward–backward splitting algorithms are inversely proportional to the norms $\|A\|_{\text{op}}$ and $\|A\|_2$, and for the wildfire data set these were $\|A\|_{\text{op}} \approx 3.30$ and $\|A\|_2 \approx 854.08$. Thus, larger stepsize parameters were permitted thanks to the Kullback–Leibler divergence term in the NPDHG algorithm, enabling a major speedup gain.

5.5. Analysis of the MaxEnt Regularization Paths and Estimated Fire Probabilities

As a final validation step for the NPDHG algorithm, we use the fitted MaxEnt models to compute the normalized mean fire probability in each grid cell for all calendar months between 1984 and 2020. The probabilities are visualized in Figures 2 and 3. In each case, we have chosen the prior distribution as a benchmark for our spatial plots of fire probability.

The spatial fire probabilities for $\alpha = \{0.95, 0.40, 0.15, 0.05\}$ are shown in Figure 2. The range of α values roughly corresponds to varying the regularization from a purely l_1 norm ($\alpha = 1$) to a purely l_2 norm ($\alpha = 0$). For each value of α , we also consider the evolution of the spatial fire probability as we vary the hyperparameter t along the regularization path, or equivalently include additional features while fitting the MaxEnt model to wildfire data. Broadly, we observe that, for a fixed value of t/t_{max} , the ratio of predicted to prior fire probability decreases in sharpness as α decreases. On the other hand, for a fixed α , decreasing t/t_{max} values enables the model to make sharper distinctions between grid cells with high and low fire probability as evidenced by the sharper contrast between the prior and predicted fire probabilities. We also note a similar pattern in Figure 3 for the MaxEnt models with non-overlapping group lasso (corresponding roughly to the elastic net case with $\alpha = 1$) and the ℓ_∞ -MaxEnt models, all of which converge to the empirical distribution quicker than any of the elastic net cases.

In Figure 4, we show the cumulative number of non-zero coefficients at fixed intervals along the regularization path for different α values. The plot helps in visualizing the t/t_{max} values at which new features are introduced in the elastic net MaxEnt model, with the dashed vertical lines indicating the first time a feature from a new group is selected. Across all α values, we find that features appear in the same order, with fire weather features being selected first, followed by topography, vegetation, human, and antecedent features. We tabulate the final set of non-zero features at the end of the regularization path for $\alpha = \{0.95, 0.05\}$ across various groups in Table 2. These selected features are in good agreement with physical models of wildfire occurrence [42,43] as well as previous statistical analyses of wildfire drivers [29–31,44].

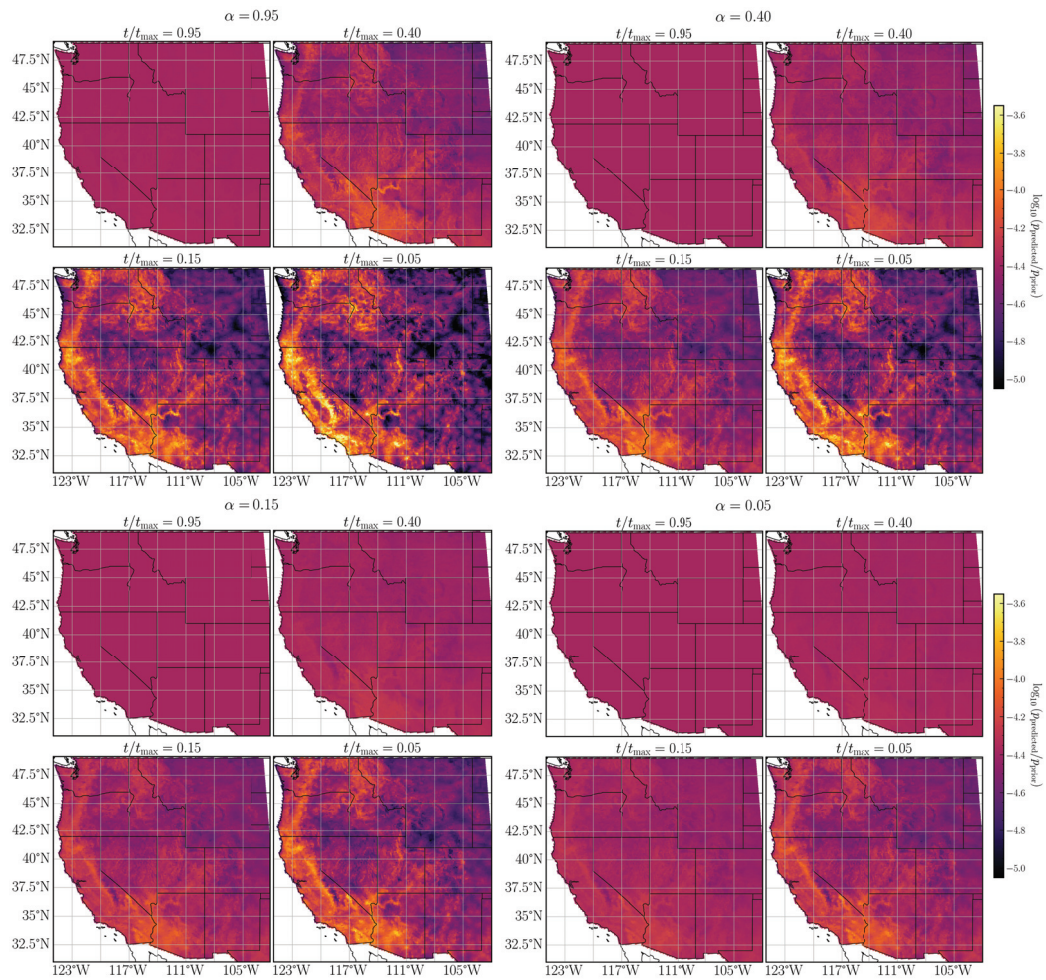


Figure 2. Spatial probability plot for different hyperparameter values with elastic net penalty parameter $\alpha = \{0.95, 0.40, 0.15, 0.05\}$.

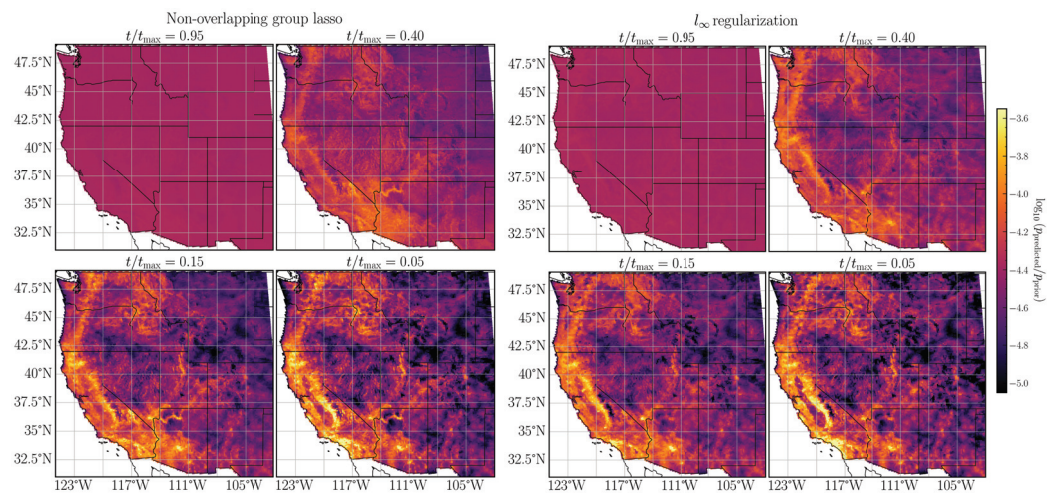


Figure 3. Same as Figure 2 but for (left) the non-overlapping group lasso with $\alpha = 1$ and (right) the l_∞ MaxEnt models, respectively.

Table 2. List of non-zero features at the end of the regularization path for two elastic net penalty parameters, organized by different feature groups. See Appendix B for additional description of the selected features.

Feature Group	$\alpha = 0.95$	$\alpha = 0.05$	
Fire weather	Tmax	Tmax	VPD
	Prec	Prec	Tmin
	Wind	Wind	VPD _{max3}
	FM1000	FM1000	VPD _{max7}
	Tmin _{max3}	Tmin _{max3}	Tmax _{max3}
	Lightning	Lightning	Tmax _{max7}
		Tmin _{max7}	SWE _{max}
Topography	Slope	Slope	Southness
	Southness		
Vegetation	Grassland	Biomass	Forest
		Shrub	
Human	Camp_dist	Camp_dist	Road_dist
	Camp_num	Camp_num	
	Urban	Urban	
	Popdensity	Popdensity	
Antecedent	AvgPrec _{2mo}	AvgPrec _{2mo}	AntPrec _{lag2}
	AvgVPD _{3mo}	AntPrec _{lag1}	

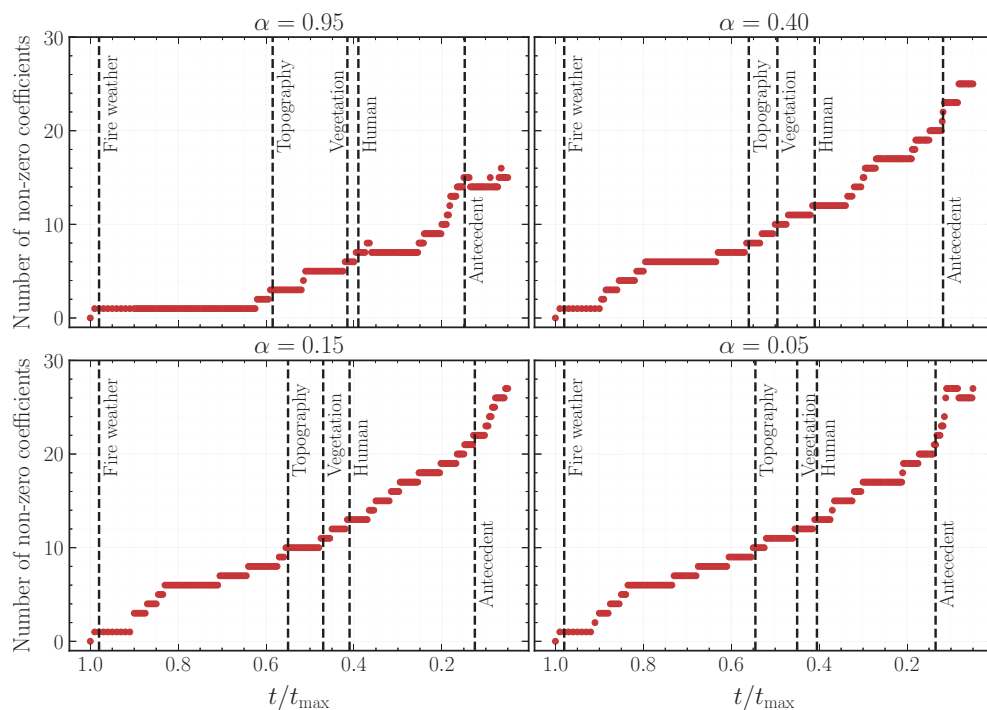


Figure 4. Number of non-zero coefficients along the regularization path plots for elastic net penalty parameter $\alpha = \{0.95, 0.40, 0.15, 0.05\}$. The dashed vertical lines highlight the $t/t_{\max}$ value at which the first feature of the group indicated by inset text is selected.

6. Discussion

In this paper, we have introduced novel first-order NPDHG algorithms that overcome the shortcomings of state-of-the-art algorithms for training large-scale, non-smooth MaxEnt models. The crucial ingredient common to our algorithms is the Kullback–Leibler divergence. Using it over the classical proximal operator makes it possible to train large-scale and non-smooth MaxEnt models much more efficiently than the state of the art. In particular, all stepsize parameters and iterations in our algorithms can be calculated on the order $O(mn)$ operations, improving on the complexity bound of $O(\min(m^2n, mn^2))$ operations for computing the optimal stepsize parameters of classical first-order optimization methods, such as the linear PDHG or forward–backward splitting methods. As a consequence, for a given tolerance level $\epsilon > 0$, our algorithms provably compute solutions using on the order of $O(mn/\sqrt{\epsilon})$ or $O(mn/\log(1/\epsilon))$ operations, the order depending on the smoothness of the MaxEnt model and which is optimal with respect to the Nesterov class of optimal first-order methods [39]. Moreover, the computational bottleneck consists of matrix–vector multiplications, which can be trivially parallelized, and so our algorithms exhibit scalable parallelism.

Finally, we have shown that the strong convexity of the Kullback–Leibler divergence with respect to the ℓ_1 norm allows for significantly larger stepsize parameters, thereby speeding up the convergence rate of our algorithms. This was, in particular, observed in Section 5, when we applied our algorithms to fit the WUMI wildfire data set [45] on several non-smooth MaxEnt models to estimate the probabilities of fire occurrences as a function of ecological features. Our algorithms outperformed the state-of-the-art forward–backward splitting and coordinate descent STRUCTMAXENT2 algorithms by at least one order of magnitude. They also yielded results that are in good agreement with physical models of wildfire occurrence [42,43] as well as previous statistical analyses of wildfire drivers [29–31,44]. Future work will explore the scalability of our algorithms for modeling daily scale wildfire probability [80]. It will also explore how the choices of different informative priors affect the MaxEnt models using prediction accuracy as a metric.

We expect our algorithms to provide efficient methods for solving non-smooth MaxEnt models that arise in large-scale machine learning applications beyond the wildfire application explored in this paper. An interesting future direction, albeit tangential to the focus of this work, would be to prove learning guarantees for the MaxEnt problem (3) for arbitrary choices of potential functions, similarly to [19,64], for the choice of ℓ_1 and ℓ_2 norms as potential functions. This would provide some means to assess the “quality” of the probability distribution estimated from the MaxEnt problem. Another direction that would be interesting and impactful is to extend our algorithms to continuous regularized MaxEnt models and prove rigorously that they work: interesting because the continuous version of the MaxEnt problem is an infinite dimensional problem, which makes this problem more technically challenging, impactful because such an algorithm would enable a much broader class of probability distributions to be used in MaxEnt modeling.

Author Contributions: Conceptualization, G.P.L., J.B. and J.D.; data curation, J.B.; formal analysis, G.P.L.; methodology, G.P.L. and J.B.; software, G.P.L. and J.B.; visualization G.P.L. and J.B.; writing—original draft G.P.L., J.B. and J.D.; writing—review & drafting G.P.L., J.B. and J.D. All authors have read and agreed to the published version of the manuscript.

Funding: G.P.L. and J.D. are supported by ONR N00014-22-1-2667. J.D. is also supported by DOE-MMICS SEA-CROGS DE-SC0023191. J.B. is supported by the Zegar Family Foundation and the National Science Foundation LEAP Science and Technology Center Award #2019625.

Institutional Review Board Statement: Not Applicable.

Data Availability Statement: The data that support the findings of this study are openly available in Zenodo at <https://zenodo.org/records/7277980> (accessed on 1 April 2024). The code used to produce the results from this project can be found on Github at <https://github.com/Gabriel-P-Langlois/maxent-paper-wildfire>.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

Appendix A. Derivation of Update (17)

We divide the derivation into two parts, first deriving an expression for the update p_{k+1} in terms of the previous iterate p_k and then deriving the explicit update for p_{k+1} as in the third line of (16).

Part 1. First, note the minimization problem

$$\min_{p \in \Delta_n} \{ \tau_k D_{\text{KL}}(p \parallel p_{\text{prior}}) - \tau_k \langle \mathbf{w}_k + \theta_k(\mathbf{w}_k - \mathbf{w}_{k-1}), \mathbb{E}_p[\Phi] \rangle + D_{\text{KL}}(p \parallel p_k) \}$$

has a unique global minimum, denoted by p_{k+1} , in the relative interior of the unit simplex Δ_n for every $k \in \mathbb{N}$. See ([38], Proposition 4.1) for details and note that Proposition 4.1 applies because the Kullback–Leibler divergence is strongly convex with respect to the ℓ_1 norm in its first argument.

Having established that the minimum p_{k+1} exists and is unique, we introduce the Karush–Kuhn–Tucker multipliers $\xi \in \mathbb{R}$ and $\mu \in \mathbb{R}^n$ to write the minimization problem for p_{k+1} in terms of an unconstrained optimization problem:

$$\min_{\substack{p \in \mathbb{R}^n \\ \xi \in \mathbb{R} \\ \mu \in \mathbb{R}^n}} \left\{ \tau_k D_{\text{KL}}(p \parallel p_{\text{prior}}) - \tau_k \langle \mathbf{w}_k + \theta_k(\mathbf{w}_k - \mathbf{w}_{k-1}), \mathbb{E}_p[\Phi] \rangle + D_{\text{KL}}(p \parallel p_k) + \xi \left(1 - \sum_{j=1}^n p(j) \right) - \langle \mu, p \rangle \right\}.$$

We can invoke the Karush–Kuhn–Tucker Theorem and use the linearity constraint qualifications to find that p_{k+1} satisfies the first-order optimality condition

$$\begin{aligned} \tau_k \log(p_{k+1}(j)/p_{\text{prior}}(j)) + \tau_k - \tau_k \langle \mathbf{w}_k + \theta_k(\mathbf{w}_k - \mathbf{w}_{k-1}), \Phi(j) \rangle \\ + \log(p_{k+1}(j)/p_k(j)) + 1 - \xi - [\mu]_j = 0 \end{aligned} \quad (\text{A1})$$

for each $j \in \{1, \dots, n\}$ [81]. Moreover, the complementary slackness condition $\langle \mu, p_{k+1} \rangle = 0$ with the fact that $p_{k+1} > 0$ for every $j \in \{1, \dots, n\}$ implies $\mu = \mathbf{0}$.

Next, we use the first-order optimality condition to compute $p_{k+1}(j)$ explicitly, rearranging (A1) in terms of $\log(p_{k+1}(j))$, using $\mu = \mathbf{0}$, and taking the exponential yield

$$p_{k+1}(j) = \left(e^{(\xi-1)/(1+\tau_k)-1} \right) \left((p_{\text{prior}}(j))^{\tau_k} p_k(j) e^{\tau_k \langle \mathbf{w}_k + \theta_k(\mathbf{w}_k - \mathbf{w}_{k-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_k)}$$

Since $\sum_{j=1}^n p_{k+1}(j) = 1$, we find that ξ satisfies the relation

$$e^{\tau_k \xi / (1+\tau_k) - 1} = \frac{1}{\sum_{j=1}^n \left((p_{\text{prior}}(j))^{\tau_k} p_k(j) e^{\tau_k \langle \mathbf{w}_k + \theta_k(\mathbf{w}_k - \mathbf{w}_{k-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_k)}}.$$

Hence,

$$p_{k+1}(j) = \frac{\left((p_{\text{prior}}(j))^{\tau_k} p_k(j) e^{\tau_k \langle \mathbf{w}_k + \theta_k(\mathbf{w}_k - \mathbf{w}_{k-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_k)}}{\sum_{j=1}^n \left((p_{\text{prior}}(j))^{\tau_k} p_k(j) e^{\tau_k \langle \mathbf{w}_k + \theta_k(\mathbf{w}_k - \mathbf{w}_{k-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_k)}}$$

for each $j \in \{1, \dots, n\}$. This yields an explicit update for p_{k+1} in terms of the previous iterate p_k .

Part 2. We now induce on k to prove that the explicit expression for the update p_{k+1} in the third line of (16) is correct. For $k = 0$, we have

$$p_0(j) = \frac{p_{\text{prior}}(j) e^{\langle \mathbf{z}_0, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle \mathbf{z}_0, \Phi(j) \rangle}}$$

and so

$$\begin{aligned} & \left((p_{\text{prior}}(j))^{\tau_0} p_0(j) e^{\tau_0 \langle w_0 + \theta_0(w_0 - w_{-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_0)} \\ &= \frac{p_{\text{prior}}(j) e^{\langle (z_0 + \tau_0(w_0 + \theta_0(w_0 - w_{-1}))) / (1+\tau_0), \Phi(j) \rangle}}{\left(\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_0, \Phi(j) \rangle} \right)^{1/(1+\tau_0)}}. \end{aligned}$$

Writing

$$z_1 = (z_0 + \tau_0(w_0 + \theta_0(w_0 - w_{-1}))) / (1 + \tau_0),$$

the term inside the exponential on the numerator simplifies to $\langle z_1, \Phi(j) \rangle$. Hence,

$$\left((p_{\text{prior}}(j))^{\tau_0} p_0(j) e^{\tau_0 \langle w_0 + \theta_0(w_0 - w_{-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_0)} = \frac{p_{\text{prior}}(j) e^{\langle z_1, \Phi(j) \rangle}}{\left(\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_0, \Phi(j) \rangle} \right)^{1/(1+\tau_0)}}.$$

It follows on substitution that

$$p_1(j) = \frac{p_{\text{prior}}(j) e^{\langle z_1, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_1, \Phi(j) \rangle}}$$

for every $j \in \{1, \dots, n\}$.

For arbitrary $k \in \mathbb{N}$, we use the induction hypothesis

$$p_k(j) = \frac{p_{\text{prior}}(j) e^{\langle z_k, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_k, \Phi(j) \rangle}}$$

to find

$$\begin{aligned} & \left((p_{\text{prior}}(j))^{\tau_k} p_k(j) e^{\tau_k \langle w_k + \theta_k(w_k - w_{k-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_k)} \\ &= \frac{p_{\text{prior}}(j) e^{\langle (z_k + \tau_k(w_k + \theta_k(w_k - w_{k-1}))) / (1+\tau_k), \Phi(j) \rangle}}{\left(\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_k, \Phi(j) \rangle} \right)^{1/(1+\tau_k)}}. \end{aligned}$$

Writing

$$z_{k+1} = (z_k + \tau_k(w_k + \theta_k(w_k - w_{k-1}))) / (1 + \tau_k),$$

the term inside the exponential on the numerator simplifies to $\langle z_{k+1}, \Phi(j) \rangle$. Hence,

$$\left((p_{\text{prior}}(j))^{\tau_k} p_k(j) e^{\tau_k \langle w_k + \theta_k(w_k - w_{k-1}), \Phi(j) \rangle} \right)^{1/(1+\tau_k)} = \frac{p_{\text{prior}}(j) e^{\langle z_{k+1}, \Phi(j) \rangle}}{\left(\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_k, \Phi(j) \rangle} \right)^{1/(1+\tau_k)}}.$$

It follows on substitution that

$$p_{k+1}(j) = \frac{p_{\text{prior}}(j) e^{\langle z_{k+1}, \Phi(j) \rangle}}{\sum_{j=1}^n p_{\text{prior}}(j) e^{\langle z_{k+1}, \Phi(j) \rangle}}$$

for every $j \in \{1, \dots, n\}$, which proves the desired result.

Appendix B. Summary of Features Extracted from the WUMI Wildfire Data Set

We provide here a summary of all the features used in our statistical analysis. The features are aggregated to a 12 km spatial resolution during data preprocessing. Considering each feature's M antecedent month average and maximum X -day running average components as distinct features, the total number of features considered in our analysis

adds up to 35. We organize the features into five groups with each group reflecting a major environmental driver of fire occurrences. These groups are also an important ingredient of the non-overlapping group lasso MaxEnt model.

Table A1. Summary table for all input features organized by group, identifier, description, spatial resolution of raw data, and source of data.

Feature Group	Identifier	Description	Resolution	Source
Fire weather	VPD	Mean vapor pressure deficit	5 km	Climgrid [82], PRISM [83]
	$VPD^{\max X}$	Maximum X-day running average of VPD; $X \in \{3, 7\}$	9 km	UCLA-ERA5 [84]
	Tmax	Daily maximum temperature	5 km	Climgrid
	$Tmax^{\max X}$	Maximum X-day running average of Tmax	9 km	UCLA-ERA5
	Tmin	Daily minimum temperature	5 km	Climgrid
	$Tmin^{\max X}$	Maximum X-day running average of Tmin	9 km	UCLA-ERA5
	Prec	Precipitation total	5 km	Climgrid
	SWE_{mean}	Mean snow water equivalent	500 m	NSIDC [85]
	SWE_{max}	Daily maximum snow water equivalent	500 m	NSIDC
	FM1000	1000 h dead fuel moisture	4 km	gridMET [86]
	$FFWI^{\max 7}$	Maximum 7-day running average of Fosberg Fire Weather Index	9 km	UCLA-ERA5
	Wind	Mean wind speed	9 km	UCLA-ERA5
	Lightning	Lightning strike density	500 m	NLDN [87,88]
Antecedent	$AntVPD_{M\text{mon}}$	Average VPD in M antecedent months; $M \in \{2, 3\}$	5 km	Climgrid
	$AntPrec_{M\text{mon}}$	Average precipitation total in M antecedent months; $M \in \{2, 4\}$	5 km	Climgrid
	$AntPrec_{\text{lag}1}$	Mean annual precipitation in lag year 1	5 km	Climgrid
	$AntPrec_{\text{lag}2}$	Mean annual precipitation in lag year 2	5 km	Climgrid
	$AvgSWE_{3\text{mon}}$	Average snow water equivalent in 3 antecedent months	500 m	NSIDC
Vegetation	Forest	Fraction of forest landcover	30 m	NLCD
	Grassland	Fraction of grassland cover	30 m	NLCD
	Shrubland	Fraction of shrubland cover	30 m	NLCD
	Biomass	Aboveground biomass map	300 m	Ref. [89]
Human	Urban	Fraction of land covered by urban areas	30 m	NLCD
	Camp_num	Mean number of camp grounds	1 km	Open source
	Camp_dist	Mean distance from nearest camp ground	1 km	Open source
	Road_dist	Mean distance from nearest highway	1 km	Open source
	Popdensity	Mean population density	1 km	SILVIS [90]
	Housedensity	Mean housing density	1 km	SILVIS
Topography	Slope	Mean slope	1 m	USGS
	Southness	Mean south-facing degree of slope	1 m	USGS

References

- Jaynes, E.T. Information theory and statistical mechanics. *Phys. Rev.* **1957**, *106*, 620. [CrossRef]
- Jaynes, E.T. Information theory and statistical mechanics. II. *Phys. Rev.* **1957**, *108*, 171. [CrossRef]
- Kapur, J.N. *Maximum-Entropy Models in Science and Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 1989.
- Gu, Y.; McCallum, A.; Towsley, D. Detecting anomalies in network traffic using maximum entropy estimation. In Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement, Berkeley, CA, USA, 19–21 October 2005; p. 32.
- Bereziński, P.; Jasiul, B.; Szpyrka, M. An entropy-based network anomaly detection method. *Entropy* **2015**, *17*, 2367–2408. [CrossRef]
- Wainwright, M.J.; Jordan, M.I. *Graphical Models, Exponential Families, and Variational Inference*; Now Publishers Inc.: Hanover, MA, USA, 2008.
- Berger, A.; Pietra, S.A.D.; Pietra, V.J.D. A maximum entropy approach to natural language processing. *Comput. Linguist.* **1996**, *22*, 39–71.
- Chen, S.F.; Rosenfeld, R. A survey of smoothing techniques for me models. *IEEE Trans. Speech Audio Process.* **2000**, *8*, 37–50. [CrossRef]

9. Pietra, S.D.; Pietra, V.D.; Lafferty, J. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 380–393. [CrossRef]
10. Malouf, R. A comparison of algorithms for maximum entropy parameter estimation. In Proceedings of the COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002), Taipei, Taiwan, 31 August 2002.
11. Ratnaparkhi, A. *Maximum Entropy Models for Natural Language Processing*; Springer: Boston, MA, USA, 2017; pp. 800–805.
12. Tsujii, J.; Kazama, J. Evaluation and extension of maximum entropy models with inequality constraints. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan, 11–12 July 2003; pp. 137–144.
13. Hayes, B.; Wilson, C. A maximum entropy model of phonotactics and phonotactic learning. *Linguist. Inq.* **2008**, *39*, 379–440. [CrossRef]
14. Kwon, T.M. *Tmc Traffic Data Automation for Mndot's Traffic Monitoring Program*; Technical Report; University of Minnesota: Twin Cities, MN, USA, 2004.
15. Muttaqin, L.A.; Murti, S.H.; Susilo, B. Maxent (maximum entropy) model for predicting prehistoric cave sites in karst area of gunung sewu, gunung kidul, yogyakarta. In *Sixth Geoinformation Science Symposium*; SPIE-International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 11311, p. 113110B.
16. Granot-Atedgi, E.; Tkačik, G.; Segev, R.; Schneidman, E. Stimulus-dependent maximum entropy models of neural population codes. *PLoS Comput. Biol.* **2013**, *9*, e1002922. [CrossRef]
17. Tkačik, G.; Marre, O.; Mora, T.; Amodei, D.; Berry, M.J., II; Bialek, W. The simplest maximum entropy model for collective behavior in a neural network. *J. Stat. Mech. Theory Exp.* **2013**, *2013*, P03011. [CrossRef]
18. Savin, C.; Tkačik, G. Maximum entropy models as a tool for building precise neural controls. *Curr. Opin. Neurobiol.* **2017**, *46*, 120–126. [CrossRef] [PubMed]
19. Dudík, M.; Phillips, S.J.; Schapire, R.E. Performance guarantees for regularized maximum entropy density estimation. In *Learning Theory*; Shawe-Taylor, J., Singer, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 472–486.
20. Dudík, M.; Phillips, S.J.; Schapire, R.E. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *J. Mach. Learn. Res.* **2007**, *8*, 1217–1260.
21. Elith, J.; Phillips, S.J.; Hastie, T.; Dudík, M.; Chee, Y.E.; Yates, C.J. A statistical explanation of maxent for ecologists. *Divers. Distrib.* **2011**, *17*, 43–57. [CrossRef]
22. Kalinski, C.E. Building Better Species Distribution Models with Machine Learning: Assessing the Role of Covariate Scale and Tuning in Maxent Models. Ph.D. Thesis, University of Southern California, Los Angeles, CA, USA, 2019.
23. Merow, C.; Smith, M.J.; Silander, J.A., Jr. A practical guide to maxent for modeling species' distributions: What it does, and why inputs and settings matter. *Ecography* **2013**, *36*, 1058–1069. [CrossRef]
24. Phillips, S.J.; Dudík, M.; Schapire, R.E. A maximum entropy approach to species distribution modeling. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 83.
25. Phillips, S.J.; Anderson, R.P.; Schapire, R.E. Maximum entropy modeling of species geographic distributions. *Ecol. Model.* **2006**, *190*, 231–259. [CrossRef]
26. Phillips, S.J.; Anderson, R.P.; Dudík, M.; Schapire, R.E.; Blair, M.E. Opening the black box: An open-source release of maxent. *Ecography* **2017**, *40*, 887–893. [CrossRef]
27. Schnase, J.L.; Carroll, M.L.; Gill, R.L.; Tamkin, G.S.; Li, J.; Strong, S.L.; Maxwell, T.P.; Aronne, M.E.; Spradlin, C.S. Toward a monte carlo approach to selecting climate variables in maxent. *PLoS ONE* **2021**, *16*, e0237208. [CrossRef]
28. Schnase, J.L.; Carroll, M.L. Automatic variable selection in ecological niche modeling: A case study using cassin's sparrow (*peucaea cassinii*). *PLoS ONE* **2022**, *17*, e0257502. [CrossRef] [PubMed]
29. Parisien, M.; Moritz, M.A. Environmental controls on the distribution of wildfire at multiple spatial scales. *Ecol. Monogr.* **2009**, *79*, 127–154. [CrossRef]
30. Chen, B.; Jin, Y.; Scaduto, E.; Moritz, M.A.; Goulden, M.L.; Randerson, J.T. Climate, fuel, and land use shaped the spatial pattern of wildfire in California's Sierra Nevada. *J. Geophys. Res. Biogeosci.* **2021**, *126*, e2020JG005786. [CrossRef]
31. Yu, Y.; Saatchi, S.S.; Walters, B.F.; Ganguly, S.; Li, S.; Hagen, S.; Melendy, L.; Nemani, R.R.; Domke, G.M.; Woodall, C.W. Carbon Pools across Conus Using the Maxent Model, 2005, 2010, 2015, 2016, and 2017. 2021. Available online: https://daac.ornl.gov/CMS/guides/CMS_CONUS_Biomass.html (accessed on 1 April 2024).
32. L'heureux, A.; Grolinger, K.; Elyamany, H.F.; Capretz, M.A.M. Machine learning with big data: Challenges and approaches. *IEEE Access* **2017**, *5*, 7776–7797. [CrossRef]
33. Fithian, W.; Hastie, T. Finite-sample equivalence in statistical models for presence-only data. *Ann. Appl. Stat.* **2013**, *7*, 1917. [CrossRef] [PubMed]
34. Darroch, J.N.; Ratcliff, D. Generalized iterative scaling for log-linear models. In *The Annals of Mathematical Statistics*; Institute of Mathematical Statistics: Waite Hill, OH, USA, 1972; pp. 1470–1480.
35. Demchenko, Y.; Grosso, P.; Laatz, C.D.; Membrey, P. Addressing big data issues in scientific data infrastructure. In Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA, 20–24 May 2013; pp. 48–55.
36. Thompson, N.C.; Greenewald, K.; Lee, K.; Manso, G.F. Deep learning's diminishing returns: The cost of improvement is becoming unsustainable. *IEEE Spectr.* **2021**, *58*, 50–55. [CrossRef]

37. Dyderski, M.K.; Paż, S.; Frelich, L.E.; Jagodziński, A.M. How much does climate change threaten european forest tree species distributions? *Glob. Chang. Biol.* **2018**, *24*, 1150–1163. [CrossRef] [PubMed]
38. Darbon, J.; Langlois, G.P. Accelerated nonlinear primal-dual hybrid gradient algorithms with applications to machine learning. *arXiv* **2021**, arXiv:2109.12222.
39. Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*; Kluwer Academic Publishers: Boston, MA, USA, 2004; Volume 87.
40. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
41. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
42. Parisien, M.; Dawe, D.A.; Miller, C.; Stockdale, C.A.; Armitage, O.B. Applications of simulation-based burn probability modelling: A review. *Int. J. Wildland Fire* **2019**, *28*, 913. [CrossRef]
43. Mezuman, K.; Tsigaridis, K.; Faluvegi, G.; Bauer, S.E. The interactive global fire module pyre (v1.0). *Geosci. Model Dev.* **2020**, *13*, 3091–3118. [CrossRef]
44. Buch, J.; Williams, A.P.; Juang, C.S.; Hansen, W.D.; Gentine, P. Smlfire1.0: A stochastic machine learning (sml) model for wildfire activity in the western united states. *Geosci. Model Dev.* **2023**, *16*, 3407–3433. [CrossRef]
45. Juang, C.S.; Williams, A.P.; Abatzoglou, J.T.; Balch, J.K.; Hurteau, M.D.; Moritz, M.A. Rapid growth of large forest fires drives the exponential response of annual forest-fire area to aridity in the western united states. *Geophys. Res. Lett.* **2022**, *49*, e2021GL097131. [CrossRef]
46. Ekeland, I.; Temam, R. *Convex Analysis and Variational Problems*; SIAM: New Delhi, India, 1999.
47. Beck, A.; Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.* **2003**, *31*, 167–175. [CrossRef]
48. Csizsár, I. Information-type measures of difference of probability distributions and indirect observation. *Stud. Sci. Math. Hung.* **1967**, *2*, 229–318.
49. Kemperman, J.H.B. On the optimum rate of transmitting information. In *Probability and Information Theory*; Springer: Berlin/Heidelberg, Germany, 1969; pp. 126–169.
50. Kullback, S. A lower bound for discrimination information in terms of variation (corresp.). *IEEE Trans. Inf. Theory* **1967**, *13*, 126–127. [CrossRef]
51. Pinsker, M.S. *Information and Information Stability of Random Variables and Processes*; Holden-Day: San Francisco, CA, USA, 1964.
52. Altun, Y.; Smola, A. Unifying divergence minimization and statistical inference via convex duality. In *International Conference on Computational Learning Theory*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 139–153.
53. Dewaskar, M.; Tosh, C.; Knoblauch, J.; Dunson, D.B. Robustifying likelihoods by optimistically re-weighting data. *arXiv* **2023**, arXiv:2303.10525.
54. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. (Stat. Methodol.)* **2005**, *67*, 301–320. [CrossRef]
55. Tay, J.K.; Narasimhan, B.; Hastie, T. Elastic net regularization paths for all generalized linear models. *J. Stat. Softw.* **2023**, *106*, 1. [CrossRef] [PubMed]
56. Foucart, S.; Rauhut, H. *Sparse Solutions of Underdetermined Systems*; Springer: New York, NY, USA, 2013; pp. 41–59.
57. Guide, M.E.; Jbilou, K.; Koukouvinos, C.; Lappa, A. Comparative study of l1 regularized logistic regression methods for variable selection. In *Communications in Statistics-Simulation and Computation*; Taylor & Francis, Inc., Philadelphia, PA, USA 2020; pp. 1–16.
58. Zanon, M.; Zamboni, G.; Susto, G.A.; McLoone, S. Sparse logistic regression: Comparison of regularization and bayesian implementations. *Algorithms* **2020**, *13*, 137. [CrossRef]
59. Lee, J.D.; Recht, B.; Srebro, N.; Tropp, J.; Salakhutdinov, R.R. Practical large-scale optimization for max-norm regularization. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2010; Volume 23.
60. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*, 2nd ed.; Data Mining, Inference, and Prediction; Springer Series in Statistics; Springer: New York, NY, USA, 2009.
61. Friedman, J.; Hastie, T.; Höfling, H.; Tibshirani, R. Pathwise coordinate optimization. *Ann. Appl. Stat.* **2007**, *1*, 302–332. [CrossRef]
62. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **2010**, *33*, 1. [CrossRef]
63. Hastie, T.; Qian, J.; Tay, K. An Introduction to Glmnet. 2021. Available online: <https://glmnet.stanford.edu/articles/glmnet.html> (accessed on 17 February 2021).
64. Cortes, C.; Kuznetsov, V.; Mohri, M.; Syed, U. Structural maxent models. In Proceedings of the 32nd International Conference on Machine Learning—Volume 37, Lille, France, 6–11 July 2015; ICML’15; pp. 391–399.
65. Chambolle, A.; Bock, T. An introduction to continuous optimization for imaging. *Acta Numer.* **2016**, *25*, 161–319. [CrossRef]
66. Daubechies, I.; Defrise, M.; De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math. J. Issued Courant Inst. Math. Sci.* **2004**, *57*, 1413–1457. [CrossRef]
67. Dudík, M.; Schapire, R.E. Maximum entropy distribution estimation with generalized regularization. In *International Conference on Computational Learning Theory*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 123–138.

68. Mann, G.; McDonald, R.; Mohri, M.; Silberman, N.; Walker, D., IV. Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2009; Volume 22.
69. Andrew, G.; Gao, J. Scalable training of l_1 -regularized log-linear models. In Proceedings of the 24th International Conference on Machine Learning, Vienna, Austria, 21–27 July 2007; pp. 33–40.
70. Lee, S.-I.; Lee, H.; Abbeel, P.; Ng, A.Y. Efficient L_1 regularized logistic regression. *AAAI* **2006**, *6*, 401–408. .
71. Yuan, G.-X.; Chang, K.-W.; Hsieh, C.-J.; Lin, C.-J. A comparison of optimization methods and software for large-scale l_1 -regularized linear classification. *J. Mach. Learn. Res.* **2010**, *11*, 3183–3234.
72. Yu, H.-F.; Huang, F.-L.; Lin, C.-J. Dual coordinate descent methods for logistic regression and maximum entropy models. *Mach. Learn.* **2011**, *85*, 41–75. [CrossRef]
73. Littell, J.S.; McKenzie, D.; Peterson, D.L.; Westerling, A.L. Climate and wildfire area burned in western u.s. ecoprovinces, 1916–2003. *Ecol. Appl.* **2009**, *19*, 1003–1021. [CrossRef]
74. McElreath, R. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018.
75. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*, 3rd ed.; Chapman & Hall/CRC Texts in Statistical Science; Taylor & Francis: Boca Raton, FL, USA, 2013.
76. Figueiredo, M.A.T.; Nowak, R.D. Wavelet-based image estimation: An empirical Bayes approach using Jeffrey’s noninformative prior. *IEEE Trans. Image Process.* **2001**, *10*, 1322–1331. [CrossRef] [PubMed]
77. Lions, P.-L.; Mercier, B. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **1979**, *16*, 964–979. [CrossRef]
78. Hiriart-Urruty, J.-B.; Lemaréchal, C. *Convex Analysis and Minimization Algorithms I: Fundamentals*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1993; Volume 305.
79. Condat, L. Fast projection onto the simplex and the l_1 ball. *Math. Program.* **2016**, *158*, 575–585. [CrossRef]
80. Keeping, T.; Harrison, S.P.; Prentice, I.C. Modelling the daily probability of wildfire occurrence in the contiguous United States. *Environ. Res. Lett.* **2024**, *19*, 024036. [CrossRef]
81. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
82. Vose, R.; Applequist, S.; Squires, M.; Durre, I.; Menne, M.; Williams, C.; Fenimore, C.; Gleason, K.; Arndt, D. Improved historical Temperature and Precipitation time series for U.S. climate divisions. *J. Appl. Meteorol. Climatol.* **2014**, *53*, 1232–1251. [CrossRef]
83. Daly, C.; Gibson, W.; Doggett, M.; Smith, J.; Taylor, G. Up-to-date monthly climate maps for the conterminous United States. In Proceedings of the 14th AMS Conference on Applied Climatology, 84th AMS Annual Meeting Combined Preprints, Seattle, WA, USA, 11 January 2004.
84. Rahimi, S.; Krantz, W.; Lin, Y.; Bass, B.; Goldenson, N.; Hall, A.; Jebo, Z.; Norris, J. Evaluation of a reanalysis-driven configuration of wrf4 over the western united states from 1980–2020. *J. Geophys. Res. Atmos.* **2022**, *127*, e2021JD035699. [CrossRef]
85. Zeng, X.; Broxton, P.; Dawson, N. Snowpack change from 1982 to 2016 over conterminous United States. *Geophys. Res. Lett.* **2018**, *45*, 12940–12947. [CrossRef]
86. Abatzoglou, J.T. Development of gridded surface meteorological data for ecological applications and modelling. *Int. J. Climatol.* **2013**, *33*, 121–131. [CrossRef]
87. Wacker, R.S.; Orville, R.E. Changes in measured lightning flash count and return stroke peak current after the 1994 U.S. National Lightning Detection Network upgrade: 1. Observations. *J. Geophys. Res. Atmos.* **1999**, *104*, 2151–2157. [CrossRef]
88. Orville, R.E.; Huffines, G.R. Cloud-to-Ground lightning in the United States: NLDN results in the first decade, 1989–98. *Mon. Weather. Rev.* **2001**, *129*, 1179–1193. [CrossRef]
89. Spawn, S.A.; Sullivan, C.C.; Lark, T.J.; Gibbs, H.K. Harmonized global maps of above and belowground biomass carbon density in the year 2010. *Sci. Data* **2020**, *7*, 112. [CrossRef] [PubMed]
90. Radeloff, V.C.; Hammer, R.B.; Stewart, S.I.; Fried, J.S.; Holcomb, S.S.; McKeefry, J.F. The Wildland-Urban Interface in the United States. *Ecol. Appl.* **2005**, *15*, 799–805. [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.

Article

Power Transmission Lines Foreign Object Intrusion Detection Method for Drone Aerial Images Based on Improved YOLOv8 Network

Hongbin Sun ^{1,*}, Qiuchen Shen ¹, Hongchang Ke ^{2,*}, Zhenyu Duan ¹ and Xi Tang ¹

¹ School of Electrical and Information Engineering, Changchun Institute of Technology, Changchun 130012, China; 20231041209@stu.ccit.edu.cn (Q.S.); 20231041212@stu.ccit.edu.cn (Z.D.); 20231041213@stu.ccit.edu.cn (X.T.)

² School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun 130012, China

* Correspondence: dq_shb@ccit.edu.cn (H.S.); rj_khc@ccit.edu.cn (H.K.); Tel.: +86-138-4310-7325 (H.S.); +86-135-0088-2674 (H.K.)

Abstract: With the continuous growth of electricity demand, the safety and stability of transmission lines have become increasingly important. To ensure the reliability of power supply, it is essential to promptly detect and address foreign object intrusions on transmission lines, such as tree branches, kites, and balloons. Addressing the issues where foreign objects can cause power outages and severe safety accidents, as well as the inefficiency, time consumption, and labor-intensiveness of traditional manual inspection methods, especially in large-scale power transmission lines, we propose an enhanced YOLOv8-based model for detecting foreign objects. This model incorporates the Swin Transformer, AFPN (Asymptotic Feature Pyramid Network), and a novel loss function, Focal SIOU, to improve both the accuracy and real-time detection of hazards. The integration of the Swin Transformer into the YOLOv8 backbone network significantly improves feature extraction capabilities. The AFPN enhances the multi-scale feature fusion process, effectively integrating information from different levels and improving detection accuracy, especially for small and occluded objects. The introduction of the Focal SIOU loss function optimizes the model's training process, enhancing its ability to handle hard-to-classify samples and uncertain predictions. This method achieves efficient automatic detection of foreign objects by comprehensively utilizing multi-level feature information and optimized label matching strategies. The dataset used in this study consists of images of foreign objects on power transmission lines provided by a power supply company in Jilin, China. These images were captured by drones, offering a comprehensive view of the transmission lines and enabling the collection of detailed data on various foreign objects. Experimental results show that the improved YOLOv8 network has high accuracy and recall rates in detecting foreign objects such as balloons, kites, and bird nests, while also possessing good real-time processing capabilities.

Keywords: transmission lines; foreign object intrusion; YOLOv8; Swin transformer; AFPN; loss function

1. Introduction

Intrusion of foreign objects into power transmission lines can not only trigger serious grid incidents such as short circuits and power outages, affecting power supply in large areas, but also potentially cause secondary disasters like fires, threatening public safety and the national economy's lifeline [1]. To ensure the stability and safety of the power system, there is an urgent need for efficient detection technologies, including drone-based inspections, to conduct real-time monitoring and rapid response for power transmission lines, preventing potential hazards [2], enhancing the grid's intelligence and reliability [3], and providing a solid guarantee for the continuous development of modern society. Images of foreign objects from different angles are shown in Figure 1.



Figure 1. Samples of foreign objects from different perspectives.

Traditional machine learning algorithms such as support vector machines (SVM), decision trees, and random forests have limited performance in object detection tasks. These algorithms rely on manually designed features, which struggle to effectively capture and process complex patterns and nonlinear relationships in images, resulting in insufficient detection accuracy and generalization capability. This performance inadequacy is particularly evident when dealing with diverse and high-dimensional data [4]. In contrast, deep learning algorithms such as convolutional neural networks (CNN) and recurrent neural networks (RNN) exhibit outstanding performance in object detection [5]. CNNs, by automatically learning multi-layer features of images, can efficiently handle large-scale and complex image data, significantly improving the accuracy and robustness of object detection [6]. Furthermore, deep learning methods such as YOLO and Faster R-CNN have demonstrated exceptional real-time performance and detection accuracy in practical applications, becoming mainstream choices in the field of object detection. Consequently, due to their superior performance and broad adaptability, deep learning algorithms have gradually replaced traditional machine learning algorithms, becoming the common technology used in object detection tasks [7]. Shi et al. [8] discussed the application of Faster R-CNN in real-time object detection, addressing the issue of efficiently detecting small objects in real-time. Wang et al. [9] discussed the application of YOLOv1 and Faster R-CNN in small object detection, addressing the application of deep learning in small object detection in remote sensing images. Min et al. [10] introduced a small object detection method based on YOLO, tackling the challenges of using YOLO for small object detection in remote sensing images. Liu et al. [11] discussed various detection algorithms, including Faster R-CNN, providing a review and comparison of different deep learning methods for general object detection, especially focusing on multiple algorithms including Faster R-CNN. Zhang et al. [12] solved the issue of insufficient performance in small object detection by introducing a multi-receptive field structure. Cheng et al. [13] conducted a review and benchmark testing on large-scale small object detection, addressing the evaluation and

improvement of small object detection algorithms, particularly the application of Cascade R-CNN. However, Faster R-CNN has issues such as insufficient real-time performance, poor detection of small objects, high computational resource consumption, limited multi-scale and deformation handling capabilities, long training time, and poor handling of occluded and overlapping objects.

Among the many deep learning methods, the YOLO (You Only Look Once) series of algorithms stand out in the field of object detection due to their efficient real-time performance and excellent detection accuracy. YOLO algorithms transform the object detection problem into a single regression problem, simultaneously predicting the category and location of objects in a single forward pass, greatly reducing computation time [14]. Compared to traditional methods and other deep learning algorithms, YOLO significantly improves detection speed while maintaining high accuracy, making it particularly prominent in practical applications [15]. Since its introduction, the YOLO series has continuously iterated from YOLOv1 to the latest YOLOv8, with each generation optimizing network structures and algorithm details, consistently enhancing small object detection effectiveness and adapting to complex scenarios [16].

Shao et al. [17] proposed an improved YOLOv8-based foreign object detection method for power transmission lines, improving the accuracy and real-time performance of foreign object detection, though performance may decline under extreme weather conditions (e.g., strong winds, heavy rain), and there are high requirements for dataset diversity and generalization capability. Tang et al. [18] proposed a YOLO-based method for detecting external damages to power transmission lines, aiming to enhance the safety and reliability of the power grid by preventing external damages, but false positives or missed detections may occur in complex backgrounds (e.g., trees, buildings), demanding high model accuracy and robustness.

Shi et al. [19] studied a privacy protection method using drones and YOLO for power line detection, improving the privacy and security of transmission tower and defect detection, though privacy protection measures may increase computational complexity, affecting real-time detection performance. Li et al. [20] proposed the DF-YOLO (Deformable YOLO) algorithm for foreign object detection on power transmission lines, significantly improving detection accuracy and robustness, but the high algorithm complexity may require more computational resources, posing challenges for real-time detection. Liu et al. [21] combined the lightweight convolutional neural network MobileNet and clustering algorithms to design the MYOLO-lite object detection algorithm, improving the speed and accuracy of forward railway track object detection through improved non-maximum suppression and loss functions, achieving an average precision of 95.74%, but it may have some issues with missed or false detections when dealing with high-speed moving objects.

Nayak et al. [22] proposed an intrusion detection system (IDS) based on deep learning, using the YOLO algorithm for object detection and achieving real-time intrusion tracking through the center of mass shift algorithm and SORT algorithm, but the complex urban environment may lead to a high false alarm rate due to background interference. Chen et al. [23] used MASK R-CNN as the base network to detect foreign objects in power networks, achieving good results in terms of speed, efficiency, and recognition accuracy compared to traditional detection methods, but the computational complexity is high, resulting in insufficient real-time performance, especially when processing large-scale data. Sevi et al. [24] proposed a deep learning method based on YOLOv8 for detecting foreign objects around railways, achieving an accuracy of 88.8% on mAP@0.5 using the RailSem19 dataset and image augmentation techniques, effectively improving railway transportation safety, but it may experience false positives or missed detections in complex backgrounds such as trees and buildings. Shan et al. [25] introduced cloud-edge technology and proposed an improved YOLOv4 network-based monitoring model for foreign object detection on transmission lines. The improved YOLOv4 network significantly enhanced feature extraction and detection accuracy, achieving a foreign object detection accuracy of 99.2% and a detection speed of 218 milliseconds, meeting real-time detection needs,

but performance may decline under extreme weather conditions such as strong winds or heavy rain.

Li et al. [26] proposed a foreign object intrusion detection method based on WiFi technology, quickly detecting and locating foreign objects in non-illuminated tunnel environments by detecting abnormal phase shifts in WiFi network radio frequency signals. Experimental results show that this method performs well in terms of detection probability and location accuracy, but the detection range and accuracy may be affected by WiFi signal strength and environmental interference. Wang et al. [27] proposed an improved YOLOv8m model, combining the Global Attention Module (GAM) and Focal-EIoU loss function to detect foreign objects on transmission lines. Experimental results show that this model improved mAP@0.5 by 2.7%, mAP0.5:0.95 by 4%, and recall rate by 6%, effectively solving detection problems in complex backgrounds of transmission lines, but the high algorithm complexity may require more computational resources. Zhao et al. [28] proposed an optimization method based on Context Information Enhancement (CIE) and Joint Heterogeneous Representation (JHR), significantly improving the detection accuracy of small and occluded objects on transmission lines, with a total mAP increase of 5.8%, but in practical applications, the complexity of the algorithm may result in high consumption of computational resources.

Based on the above proposed methods, there may be deficiencies in detection efficiency and high algorithm complexity. In some methods, detection accuracy in complex scenes may still be insufficient. To address these issues, this paper proposes an improved YOLOv8 algorithm by introducing the Swin Transformer and Adaptive Feature Pyramid Network (AFPN) and replacing the loss function with Focal-SIoU to improve the accuracy and speed of foreign object detection on transmission lines. The contributions of this paper are as follows:

- (1) Swin Transformer is adopted as the high-resolution feature extraction layer to enhance the feature extraction capability of small objects.
- (2) An Adaptive Feature Pyramid Network (AFPN) is designed, combining multi-scale features to improve the model's detection capability at different scales.
- (3) The Focal-SIoU loss function is used to balance the distribution of positive and negative samples, thereby accelerating algorithm convergence and improving detection accuracy.

Experimental results show that the improved YOLOv8 model improved mAP@0.5 by 1.5%, accuracy by 3.6%, and recall rate by 5.2%, effectively solving detection problems in complex backgrounds of transmission lines.

2. YOLOv8 Network

2.1. Principle of YOLOv8 Algorithm

According to the provided YOLOv8 structure diagram, the architecture of YOLOv8 consists of three parts: the backbone network (Backbone), the feature fusion network (Neck), and the detection head (Head). The input image first passes through a series of convolutional layers and pooling layers to extract basic features. After each convolutional layer (C3F, C2F), there may be a pooling layer to reduce the resolution of the feature map and improve computational efficiency. Finally, multi-scale features are fused through the SPPCSPC layer. The features extracted by the Backbone are fused through a feature pyramid (FPN, PAN) for multi-scale integration. Upsampling and downsampling (CSPC, C2F) operations facilitate information transfer between different scale features. The feature fusion part concatenates features of different scales to form richer feature representations [29]. The detection head performs object detection based on the fused features, including classification and bounding box regression. The feature maps at each scale are processed through convolutional layers to generate the final detection results. YOLOv8 introduces spatial depth convolution (SPD-Conv) and large selective kernel (LSK) attention mechanisms to enhance feature extraction capabilities. The smooth intersection over union (SIoU) loss function improves detection accuracy and accelerates model convergence. These improve-

ments allow YOLOv8 to significantly enhance detection performance while maintaining real-time capabilities. The YOLOv8 network architecture is shown in Figure 2.

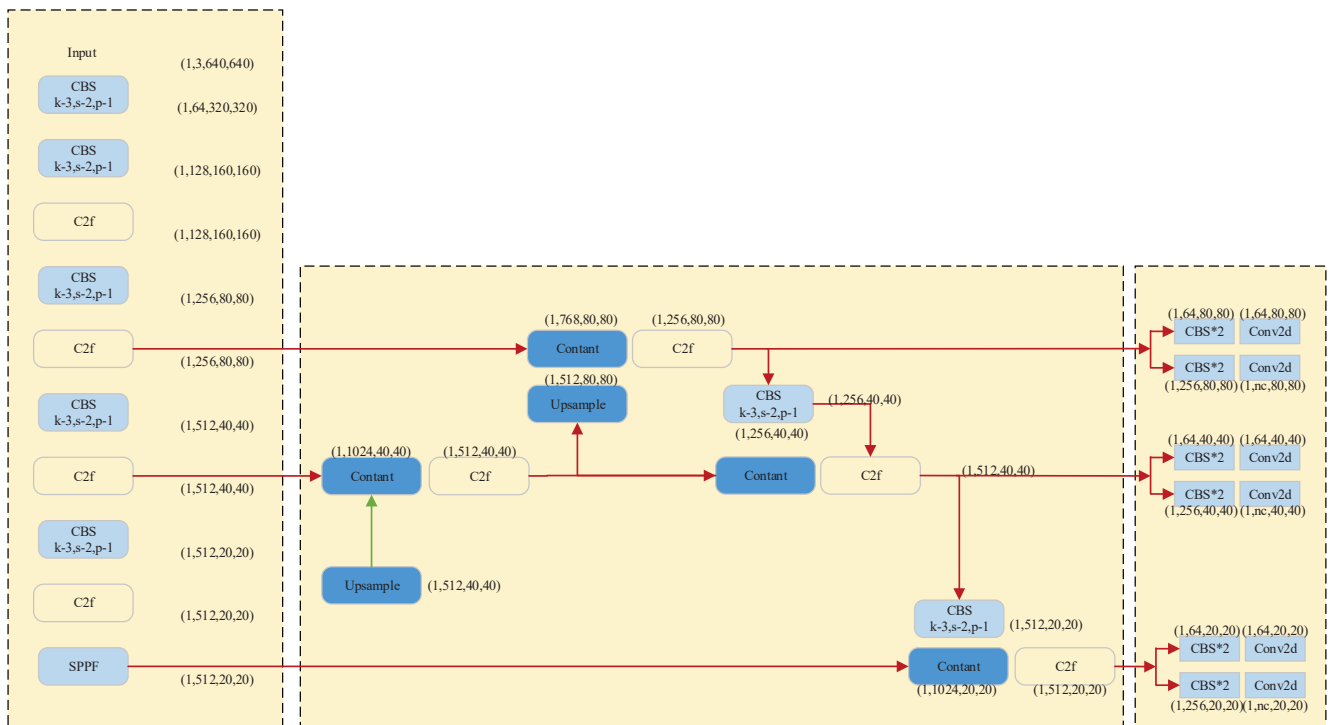


Figure 2. The YOLOv8 network architecture.

2.2. YOLOv8 Backbone Network

The backbone network of YOLOv8 is one of its core components, responsible for extracting multi-level features from the input image. The input image, sized $640 \times 640 \times 3$, first passes through a convolutional layer and a pooling layer, which initially extracts the low-level features of the image, outputting a feature map of size $320 \times 320 \times 32$ [30]. Next, it enters multiple C2F modules. These modules consist of multiple convolutional layers and feature fusion layers, gradually reducing the resolution of the feature map and increasing the number of channels to extract higher-level features [31]. For example, the first C2F module outputs a size of $160 \times 160 \times 64$, the second C2F module outputs $80 \times 80 \times 128$, the third C2F module outputs $40 \times 40 \times 256$, and the fourth C2F module outputs $20 \times 20 \times 512$. After the last C2F module, an SPP layer is used for multi-scale feature fusion, capturing contextual information at different scales, outputting a size of $20 \times 20 \times 1024$. The backbone network of YOLOv8 constructs an efficient and accurate feature extraction module through multi-level convolution and feature fusion, employing techniques such as deep convolution and large kernel convolution to enhance feature extraction efficiency and representation capability. This ensures that the model maintains high performance while having better real-time performance. Through these designs, the backbone network of YOLOv8 enhances the detection capability of small and large targets, improving detection accuracy, and speed [32]. The training flowchart of the YOLOv8 network is shown in Figure 3.

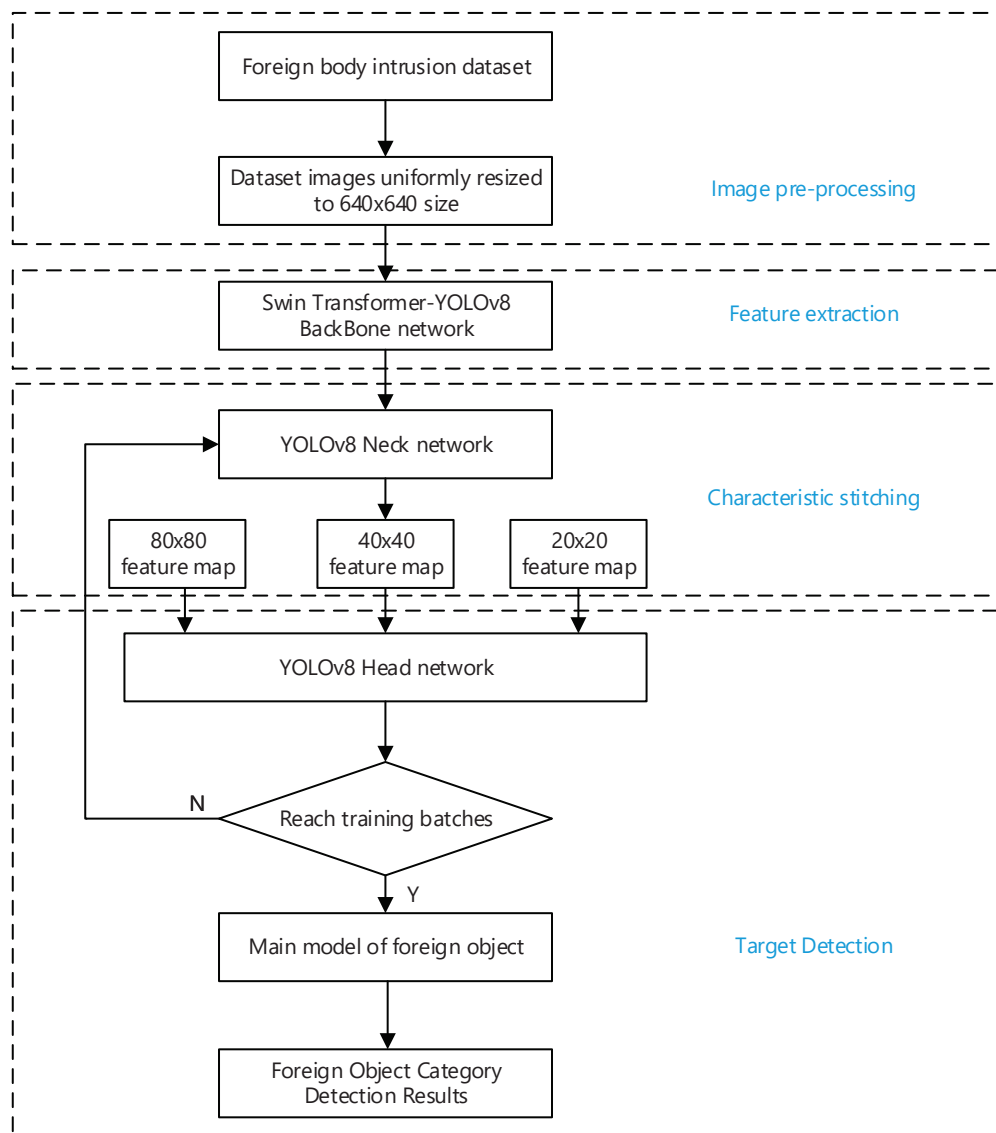


Figure 3. The training flowchart of the YOLOv8 network is shown in.

3. Improvements

3.1. Swin Transformer Overall Framework

To enhance the performance of YOLOv8 in detecting foreign objects on transmission lines, we opted to replace the backbone network of YOLOv8 with Swin Transformer. This choice is based on its remarkable performance advantages in vision tasks [33]. Swin Transformer introduces hierarchical feature representation and a sliding window mechanism, which effectively capture both local and global features in images, thereby improving detection accuracy. Compared to traditional convolutional neural networks, Swin Transformer exhibits stronger feature extraction capabilities, particularly excelling in handling complex backgrounds and small object detection [34]. Moreover, the multi-scale feature extraction mechanism of Swin Transformer allows it to be more flexible and efficient in dealing with objects of different scales. By integrating Swin Transformer with YOLOv8's single-stage detection architecture, the model's detection performance and generalization ability can be significantly enhanced. The structure of Swin Transformer is shown in Figure 4a.

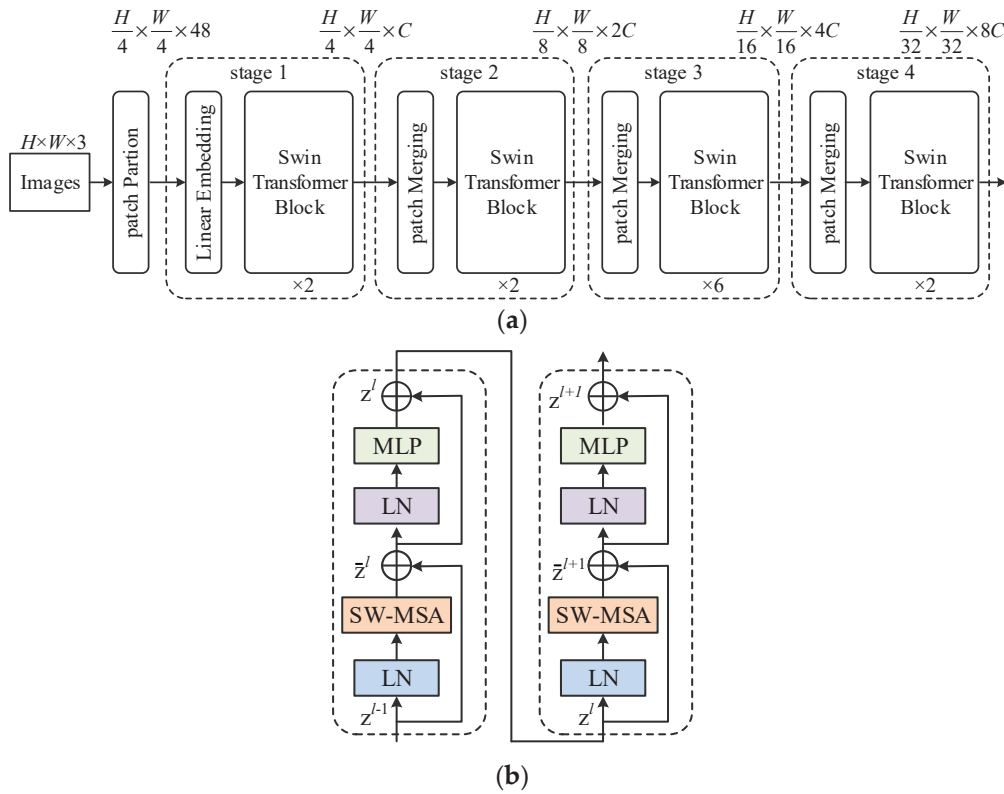


Figure 4. (a) The structure of Swin Transformer (Swin-T). (b) Two successive Swin Transformer blocks.

The architecture of Swin Transformer is outlined in Figure 4a, showcasing the small version (Swin-T). Initially, it partitions the input RGB image into non-overlapping patches via a slicing module, treating each patch as a “token” with features set to the concatenated original pixel RGB values. In our implementation, we use a patch size of 4×4 , resulting in each patch having a feature dimension of $4 \times 4 \times 3 = 48$. A linear embedding layer is then applied to these raw value features, projecting them into an arbitrary dimension (denoted as C). Several Transformer blocks (Swin Transformer blocks) with modified self-attention computation are applied to these patch tokens. These blocks maintain the number of tokens ($H/4 \times W/4$) and, together with the linear embedding layer, constitute “Stage 1”. As the network deepens, the number of tokens is reduced through patch merging layers to generate hierarchical representations. The first patch merging layer concatenates the features of 2×2 neighboring patches and applies a linear layer on the concatenated features in $4C$ dimensions. This reduces the number of tokens by a factor of $2 \times 2 = 4$ ($2 \times$ downsampling resolution), with the output dimension set to $2C$. Swin Transformer blocks are then applied to transform these features, reducing the resolution to $H/8 \times W/8$. This feature transformation after the first patch merge is termed “Stage 2”. The process repeats, sequentially reducing the resolution to $H/16 \times W/16$ and $H/32 \times W/32$, referred to as “Stage 3” and “Stage 4”, respectively. These stages collectively produce a hierarchical representation [35].

Swin Transformer constructs the Swin Transformer blocks by replacing the standard multi-head self-attention (MSA) module in the Transformer blocks with a shifted window-based module, keeping the other layers unchanged [36]. As shown in Figure 4b, each Swin Transformer block consists of a shifted window-based MSA module followed by a two-layer MLP with GELU non-linear activation. A LayerNorm (LN) layer is applied before each MSA module and each MLP, and a residual connection is applied after each module.

3.2. Improved Progressive Feature Pyramid Network (AFPN)

3.2.1. Traditional Feature Fusion

Traditional feature fusion methods in convolutional neural networks (CNNs) are typically implemented through Feature Pyramid Networks (FPN) or skip connections [37]. The core idea of these methods is to combine feature maps from different levels to capture information at various scales, thereby enhancing detection accuracy and robustness. Common traditional feature fusion methods include Feature Pyramid Networks (FPN) and skip connections. FPN enhances higher-level feature maps via a top-down pathway while integrating lower-level feature maps through lateral connections. This approach combines low-resolution semantic features and high-resolution detailed features at each level [38]. Skip connections directly combine feature maps from earlier layers with those from later layers, preserving more original information and reducing information loss [39]. The structure of the Feature Pyramid Network is illustrated in Figure 5.

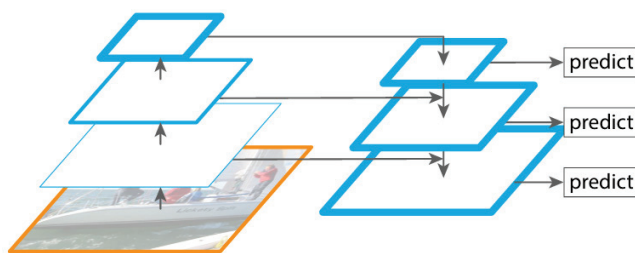


Figure 5. Feature Pyramid Network. In this figure, feature maps are indicated by blue outlines and thicker outlines denote semantically stronger features.

3.2.2. Asymptotic Feature Pyramid Network (AFPN)

Multi-scale features are crucial for encoding objects with varying scales in object detection tasks. A common strategy for multi-scale feature extraction is using classical top-down and bottom-up Feature Pyramid Networks (FPN). However, these methods suffer from feature information loss or degradation, affecting the fusion effectiveness between non-adjacent levels. This paper proposes a Asymptotic Feature Pyramid Network (AFPN) to support direct interaction between non-adjacent levels. AFPN progressively integrates high-level features into the fusion process by merging two adjacent low-level features. This approach avoids large semantic gaps between non-adjacent levels [40]. Considering the potential conflict of multi-object information during feature fusion at each spatial location, an adaptive spatial fusion operation is further adopted to mitigate these inconsistencies. We integrate the proposed AFPN into both two-stage and single-stage object detection frameworks and evaluate it on the MS-COCO 2017 validation and test sets.

Like many FPN-based object detection methods, features are extracted from different levels of the backbone before feature fusion. [41] Following the design of the Faster R-CNN framework, the last feature layer from each level of the backbone is extracted, resulting in a set of multi-scale features denoted as $\{C2, C3, C4, C5\}$. To perform feature fusion, low-level features $C2$ and $C3$ are first fed into the feature pyramid network, followed by the addition of $C4$, and finally, $C5$. After the feature fusion step, a set of multi-scale features $\{P2, P3, P4, P5\}$ is produced. For experiments conducted on the Faster R-CNN framework, a convolution with a stride of 2 is applied to $P5$, followed by a convolution with a stride of 1 to generate $P6$, ensuring uniform output. The final set of multi-scale features is $\{P2, P3, P4, P5, P6\}$, with corresponding feature strides of $\{4, 8, 16, 32, 64\}$ pixels. It should be noted that YOLO only feeds $\{C3, C4, C5\}$, excluding $C2$, into the feature pyramid network, which generates the outputs $\{P3, P4, P5\}$.

The architecture of the proposed AFPN is shown in Figure 6. During the bottom-up feature extraction process of the Backbone network, AFPN progressively integrates low-level, high-level, and top-level features [42]. Specifically, AFPN initially fuses low-level features, then integrates deeper features, and finally fuses the highest-level features, which

are the most abstract. The semantic gap between non-adjacent hierarchical features is larger than that between adjacent hierarchical features, especially between the bottom and top features. This directly leads to poorer fusion performance of non-adjacent hierarchical features. Therefore, directly using C2, C3, C4, and C5 for feature fusion is unreasonable. Due to the progressive nature of AFPN's architecture, the semantic information of different level features becomes closer during the progressive fusion process, thereby alleviating the aforementioned issue. For example, the feature fusion between C2 and C3 reduces their semantic gap. Since C3 and C4 are adjacent hierarchical features, the semantic gap between C2 and C4 is also reduced.

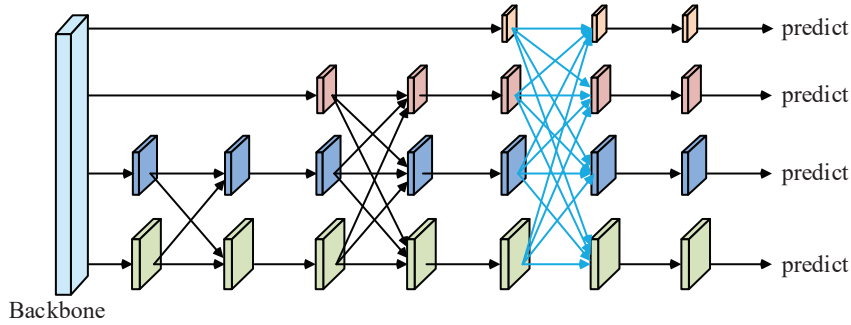


Figure 6. The architecture of the proposed Asymptotic Feature Pyramid Network (AFPN). In the initial stage, two low-level features are fused first, followed by the integration of high-level features, and finally, the top-level features are fused.

To align dimensions and prepare for feature fusion, the authors use 1×1 convolution and bilinear interpolation for upsampling. On the other hand, the authors perform downsampling using different convolution kernels and strides according to the required downsampling rate. For instance, they apply 2×2 convolution with a stride of 2 for $2 \times$ downsampling, 4×4 convolution with a stride of 4 for $4 \times$ downsampling, and 8×8 convolution with a stride of 8 for $8 \times$ downsampling. After feature fusion, the authors continue to learn features using 4 residual units, similar to ResNet. Each residual unit includes two 3×3 convolutions. Since YOLO uses only 3 levels of features, there is no $8 \times$ upsampling and $8 \times$ downsampling.

3.3. Improved Loss Function

In object detection tasks, the design of the loss function is crucial for model performance. YOLOv8 introduces an improved loss function that combines multiple strategies to enhance the model's robustness and accuracy in different scenarios. First, YOLOv8 adopts IoU (Intersection over Union) as the basis for the regression loss [43]. IoU measures the overlap between the predicted bounding box and the ground truth bounding box, defined as the area of their intersection divided by the area of their union, as shown in Equation (1):

$$IoU = \frac{B \cap B_{gt}}{B \cup B_{gt}} \quad (1)$$

However, IoU cannot reflect distance information when the bounding boxes do not overlap. Therefore, YOLOv8 introduces GIoU (Generalized IoU) and DIoU (Distance IoU) to better capture the overlap and distance relationships between target boxes. The formula for GIoU is shown in Equation (2):

$$GIoU = IoU - \frac{\left| \frac{C}{B \cup B_{gt}} \right|}{|C|} \quad (2)$$

where C is the smallest enclosing box that can contain both the predicted box and the ground truth box.

The formula for DIoU is shown in Equation (3):

$$DIoU = IoU - \frac{\rho^2(b, b_{gt})}{c^2} \quad (3)$$

where ρ is the Euclidean distance between the centers of the predicted box and the ground truth box, and c is the diagonal length of the smallest enclosing box. These improvements address the shortcomings of IoU to some extent.

Secondly, YOLOv8 employs an improved Focal Loss to address the issue of class imbalance. Focal Loss assigns greater weight to hard-to-classify samples, reducing the contribution of easily classified samples to the loss. The formula for Focal Loss is shown in Equation (4):

$$L_{FL} = -\alpha(1 - p_t)^\gamma \log(p_t) \quad (4)$$

where α is the balancing factor, γ is the modulation factor, and p_t represents the model's predicted probability for the positive class. When p_t is large, $(1 - p_t)^\gamma$ approaches zero, thereby reducing the weight of easily classified samples. By adjusting γ , the contribution ratio of hard and easy samples to the loss can be controlled.

Thirdly, confidence loss is used to measure whether the predicted box contains the target. YOLOv8 introduces Focal Loss in the confidence loss to reduce the impact of negative samples on the loss. The formula for this is shown in Equation (5):

$$L_{cof} = -\sum_{i=1}^N [a(1 - p_i)^\gamma \log(p_i) + (1 - \alpha)p_i^\gamma \log(1 - p_i)] \quad (5)$$

where p_i is the confidence of the i -th sample, and the confidence of positive and negative samples are represented by p_i and $(1 - p_i)$ respectively. By using Focal Loss to reduce the impact of negative samples, the accuracy of confidence prediction is improved.

The total loss is the weighted sum of the aforementioned components. The total loss function in YOLOv8 is given by Equation (6):

$$L = \lambda_{GIoU}L_{GIoU} + \lambda_{DIoU}L_{DIoU} + \lambda_{cls}L_{FL} + \lambda_{conf}L_{conf} \quad (6)$$

where λ_{GIoU} , λ_{DIoU} , λ_{cls} , and λ_{conf} are the weight hyperparameters for each loss component. Experimental results on standard datasets show that the improved loss function significantly enhances the object detection performance of YOLOv8, particularly in small object detection and handling complex backgrounds.

Therefore, in this paper, we adopt Focal SIoU (Smooth Intersection over Union with Focal Loss), a loss function that combines Smooth IoU and Focal Loss to enhance the performance of object detection models in handling class imbalance and bounding box regression. This loss function, through smooth IoU calculation and the weight adjustment of Focal Loss, better addresses hard and uncertain samples in object detection.

Focal SIoU combines Smooth IoU and Focal Loss by incorporating the concept of Focal Loss into the bounding box regression loss, thus optimizing the overlap of bounding boxes while paying more attention to difficult-to-locate targets and uncertain predictions. The calculation formula for Focal SIoU is given by Equation (7):

$$L_{FocalSIoU} = (1 - SIoU)^\gamma \cdot \log(SIoU) \quad (7)$$

where $SIoU$ is the result of the smooth IoU calculation, and γ is the modulation factor in Focal Loss, used to adjust the contribution of samples with different overlap degrees to the loss.

4. Experimental Analysis

4.1. Dataset

Foreign objects on power transmission lines include bird nests, kites, and balloons. The dataset used in this study is entirely sourced from a power supply bureau in Jilin

Province, China. A total of 1108 images were selected for training. The dataset was partitioned with a ratio of 9:1:1 to create the training set, validation set, and test set. To ensure a robust evaluation of the model's generalization ability, we randomly split the dataset into training, validation, and test sets in a ratio of 9:1:1. The random split ensures that the data distribution in each subset (training, validation, and test) is consistent with the overall dataset distribution, preventing performance instability due to data bias. By using random sampling, we ensured that the data in the training set, validation set, and test set are mutually exclusive, with no overlap, thus guaranteeing the fairness and independence of the model evaluation. Moreover, this strategy ensures that each subset represents the diversity and complexity of the entire dataset, making the model's performance on each subset more reliable.

We did not employ the K-fold cross-validation technique primarily due to computational resource limitations and practical application considerations. Although K-fold cross-validation can provide a more comprehensive model evaluation, its computational cost is high when dealing with large-scale image datasets. We chose a simpler and effective random split method to better simulate real-world application scenarios and more accurately evaluate the model's practical performance. The categories and quantities in the dataset are shown in Table 1.



Table 1. Categories and quantities in the dataset.

<i>Class</i>	<i>Number</i>
Nest	364
Kite	374
Balloon	370

4.2. Experimental Setup

During the experiments, a high demand was placed on the computer's memory and GPU performance due to extensive image processing requirements. The experiments were conducted on a platform with a 13th Gen Intel Core i5-13400 processor, NVIDIA GeForce RTX 4060 Ti (16 GB), and 16 GB RAM, running on the Windows 10 operating system. We used PyCharm as the deep learning software platform, with PyTorch as the deep learning framework, Anaconda3 as the integrated environment, running on Python 3.8, and CUDA 12.1 for acceleration. The experimental setup is detailed in Table 2.

Table 2. Experimental setup.

<i>Parameter</i>	<i>Setup</i>
PyTorch	2.3.0
Python	3.8
GPU	NVIDIA GeForce RTX 4060 Ti (16 GB)
CPU	13th Gen Intel Core i5-13400
RAM	16 GB
CUDA	12.1

Before conducting the experiments, multiple trials were performed on the original YOLOv8 model. Finally, the hyperparameters applied in all experiments were configured uniformly, as shown in Table 3. For example, the initial learning rate was empirically set to 0.01, with the number of iterations set to 200. To expedite the training process and accommodate hardware capabilities, the batch size was set to 32 images. All images were resized to 640×640 .

Table 3. Hyperparameter configurations.

<i>Parameter</i>	<i>Setup</i>
Initial Learning Rate	0.001
Final Learning Rate	0.001
Epoch	200
Momentum	0.937
Patience	100
IoU	0.5
Weight Decay	0.005
Image Size	640×640

The hyperparameter configurations for all training sessions are detailed in Table 3.

4.3. Evaluation Criteria

In this experiment, evaluating the performance of the YOLOv8 model requires considering various factors, including speed, accuracy, applicability, robustness, and cost. These factors are weighed differently depending on the usage scenarios. This study focuses on assessing the accuracy of the improved YOLOv8 model on the URPC series datasets. We measure the model's performance by calculating and comparing the Average Precision (AP) for each category and the mean Average Precision (mAP). Additionally, we analyze the impact of Floating Point Operations (FLOPs) and the number of parameters (Para) on the model's performance to verify the superiority of the improved YOLOv8 model.

The following formulas summarize the key metrics for evaluating the performance of object detection algorithms, including precision (P), recall (R), average precision (AP), mean average precision (mAP), and intersection over union (IoU). Precision and recall measure detection accuracy and coverage, respectively, while AP and mAP assess overall performance, and IoU evaluates the overlap between predicted and ground truth bounding boxes, as shown in Formulas (8)–(12).

$$P = \frac{TP}{TP + FP} \quad (8)$$

$$R = \frac{TP}{TP + FN} \quad (9)$$

$$AP = \int_0^1 P(R) dR \times 100\% (i = 0, 1, 2, \dots) \quad (10)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (11)$$

$$IoU = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \quad (12)$$

4.4. Comparative Experiments with Other Algorithms

4.4.1. Effect of Swin Transformer on YOLOv8

To validate the significant improvements of our enhancements compared to other algorithms, we used Faster R-CNN, SSD, YOLOv3, YOLOv5 series models, YOLOv8 series models, and our optimized model. The performance capabilities of each model parameter are shown in Table 4.

Table 4. Performance capabilities of various models.

Model	AP (%)			FPS (G)	Params (M)	mAP@0.5 (%)
	Kite	Balloon	Nest			
Faster R-CNN	73.2	85.4	77.5	38.7	129	78.7
SSD	80.3	78.2	83.6	46.1	34.3	80.6
YOLOv3	81.2	78.3	85.4	45.5	61.9	81.6
YOLOv5	81.7	84.3	85.2	109.8	24.69	83.7
YOLOv5m	94.3	86.2	79.8	89.9	21	86.7
YOLOv8	83.2	92.1	91.2	150.7	28.40	88.8
YOLOv8s	91.6	82.3	87.7	130.7	11.20	88.5
Ours Model	90.3	92.7	95.6	181.4	23.21	92.8

Based on the data in Table 4, the following conclusions can be drawn: When comparing the performance of multiple object detection models, it is evident that the YOLOv8 model with the Swin Transformer as the backbone network performs exceptionally well. In terms of AP (Average Precision), the Swin Transformer + YOLOv8 model achieved 90.3%, 92.7%, and 95.6% for kite, balloon, and bird nest detection, respectively, significantly surpassing other models. Additionally, its mAP@0.5 reached 92.8%, indicating higher average detection precision across multiple categories.

Furthermore, although the Swin Transformer + YOLOv8 has a larger number of parameters (51.3M) and FLOPs (29.75G), its outstanding detection performance demonstrates that introducing a more powerful backbone network can significantly enhance object detection accuracy and robustness. Among all the compared models, YOLOv5m and YOLOv8s also performed well, but still slightly lagged behind the improved YOLOv8 with Swin Transformer in overall performance.

Thus, it can be concluded that by integrating advanced backbone networks like Swin Transformer, YOLOv8 showcases higher accuracy and comprehensive performance in object detection tasks, despite the increased computational resource requirements.

Figure 7 shows the performance of different object detection models on the Foreign Object Intrusion Dataset, measured by Average Precision (AP %). The figure compares the detection effectiveness of various models on three types of objects: kites, balloons, and nests. It is evident that the Swin Transformer + YOLOv8 model achieves nearly or exactly 100% average precision across all object categories, demonstrating the best performance. In contrast, the average precision of Faster R-CNN, SSD, and the YOLO series models is slightly lower but still maintains a high level, approximately between 80% and 90%.

This indicates that the Swin Transformer + YOLOv8 model not only maintains high precision but also more effectively detects multiple types of objects, showcasing its advantages in complex object detection tasks.

Figure 8 compares the computational complexity (measured in FPS) of different object detection models. The models include Faster R-CNN, SSD, YOLOv3, YOLOv5, YOLOv5m, YOLOv8, YOLOv8s, and “Ours Model”. From the chart, we can see that “Ours Model” has the highest computational complexity at 181.4 FPS, while Faster R-CNN has the lowest at 38.7 FPS. The other models have the following computational complexities: SSD 46.1 FPS,

YOLOv3 45.5 FPS, YOLOv5 109.8 FPS, YOLOv5m 89.9 FPS, YOLOv8 150.7 FPS, and YOLOv8s 130.7 FPS. This chart provides a clear visual comparison of the computational complexity of each model, aiding in the selection of the most suitable model for specific application scenarios.

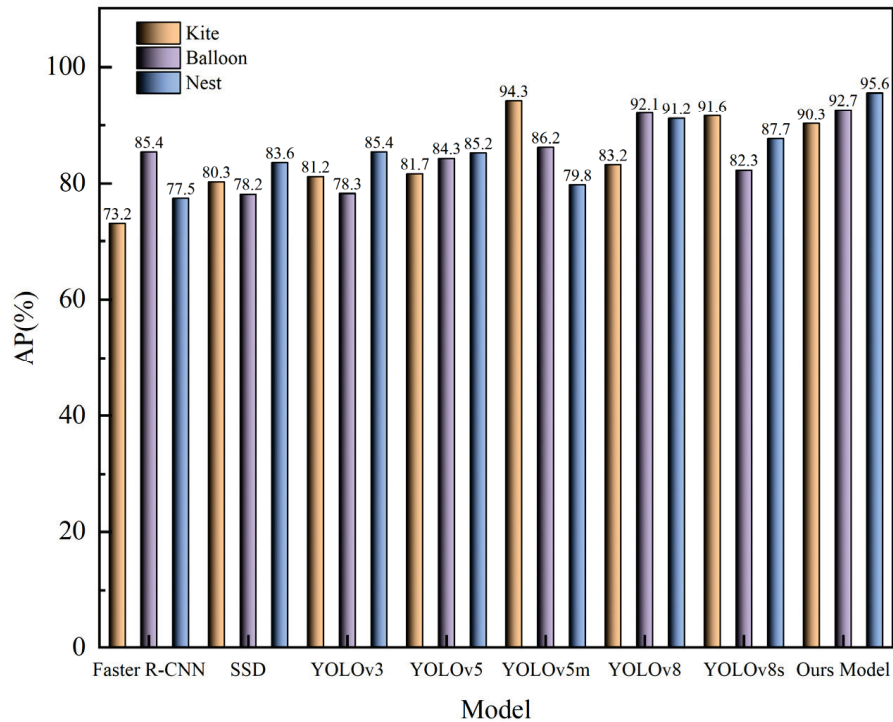


Figure 7. Performance of different models under foreign object intrusion.

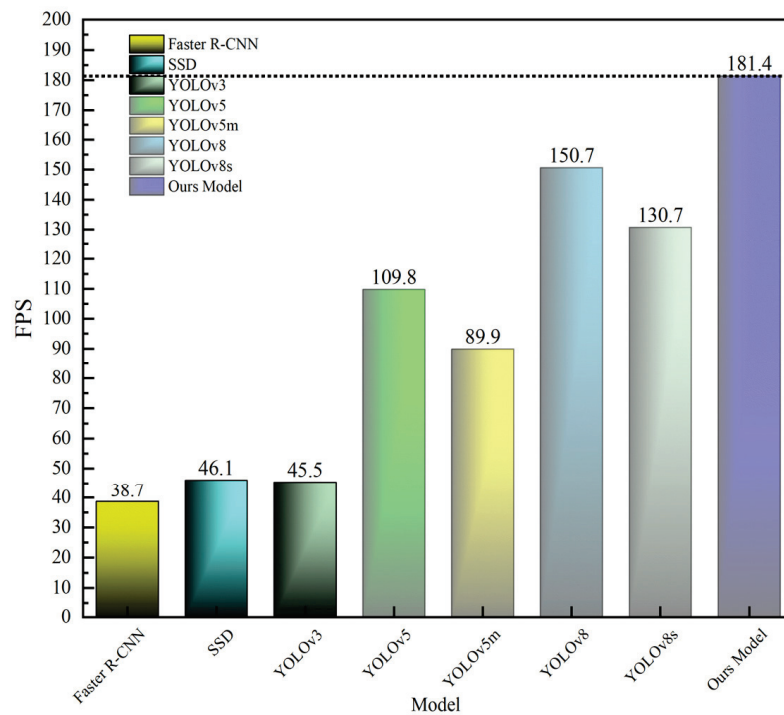


Figure 8. FPS values of different models.

Figure 9 presents a comparison of different object detection models in terms of the number of parameters (Params, in millions). The figure shows significant differences in parameter counts across the models.

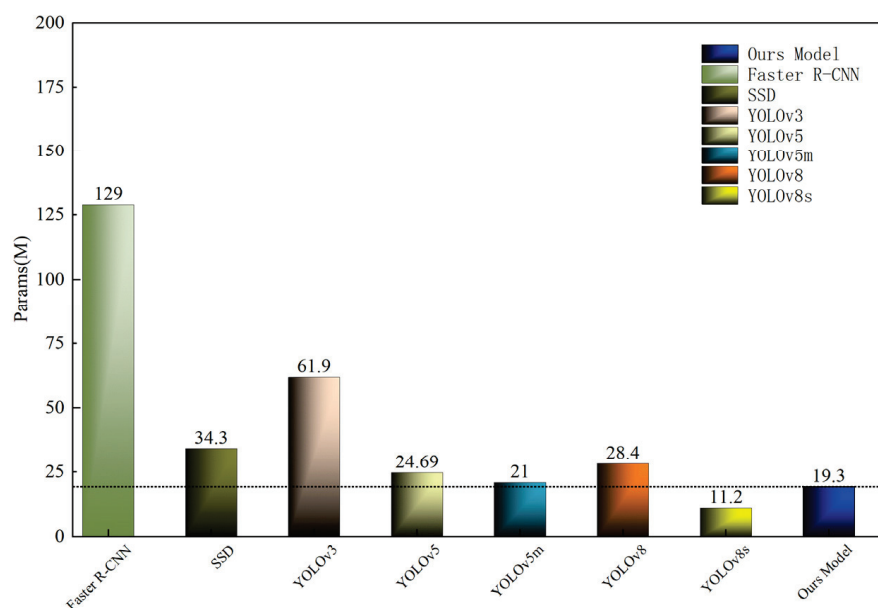


Figure 9. Params values of different models.

Specifically, Faster R-CNN has the highest parameter count, reaching 129 million. This indicates that the model is highly complex, likely requiring more computational resources and storage space, and may perform better on large datasets. The SSD model has 34.3 million parameters, fewer than Faster R-CNN, suggesting that it is relatively lightweight and may perform well in applications with high real-time requirements.

The YOLO series models exhibit variations in parameter counts across different versions. The YOLOv3 model has 61.9 million parameters, indicating a balance between performance and computational complexity. YOLOv5 has 24.69 million parameters, and YOLOv5m has 21 million parameters, demonstrating further optimization in parameter efficiency while maintaining high performance. YOLOv8 and YOLOv8s have 28.4 million and 11.2 million parameters, respectively, with YOLOv8s having the fewest parameters, showcasing significant optimization for lightweight design, making it highly suitable for embedded devices or resource-constrained environments.

Finally, our model has 19.3 million parameters, indicating a focus on reducing parameter count while potentially maintaining high performance, which is advantageous for applications requiring efficiency and low resource usage.

Figure 10 shows the mAP@0.5 (%) of different object detection models during the training process as a function of Epochs. It can be observed that our model consistently performs exceptionally well throughout the training process, quickly reaching and maintaining a high mAP value close to 0.9, significantly higher than other models.

Faster R-CNN and SSD exhibit slower mAP growth, eventually stabilizing around 0.7. While YOLOv3, YOLOv5, YOLOv5m, YOLOv8, and YOLOv8s demonstrate faster growth in the initial stages of training, their final mAP values remain lower than our model's.

Overall, our model shows significant advantages in both training speed and final accuracy, highlighting its superior performance in object detection tasks.

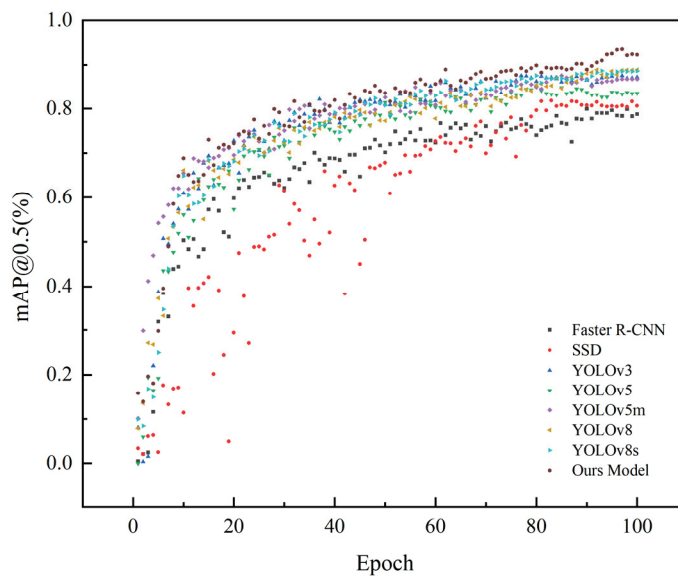


Figure 10. mAP values of different models.

4.4.2. Effect of AFPN on YOLOv8

AFPN (Adaptive Feature Pyramid Network) is an advanced feature fusion method that adaptively combines multi-scale feature information, enhancing the model's ability to detect multi-scale objects. By introducing AFPN into YOLOv8, we aim to significantly improve the model's detection accuracy in complex scenes, particularly in small object detection and occluded object detection.

In this study, we conducted a series of experiments comparing the performance of R-CNN, SPP, YOLOv3-tiny, YOLOv5m, YOLOv8s, and the YOLOv8 model with AFPN on multiple standard datasets to evaluate the effectiveness of AFPN. For this experiment, we used a dataset provided by a power supply bureau in Liaoning Province, consisting of 1895 photos. The foreign objects in these photos include trash, ribbons, sunshades, and bird nests.

We will use R-CNN, SPP, YOLOv3-tiny, YOLOv5m, YOLOv8s, and our proposed model to observe their performance on the dataset provided by the Jilin City power supply company, as shown in Table 5.

Table 5. Performance of our model and other models on the dataset.

Model	AP (%)				mAP@0.5 (%)
	Trash	Ribbon	Shade Net	Nest	
R-CNN	78.2	75.1	73.4	76.0	75.6
SPP	80.5	77.8	75.9	78.2	78.1
YOLOv3-tiny	82.3	79.6	77.5	80.1	79.8
YOLOv5m	85.4	81.2	79.8	82.7	82.1
YOLOv8s	87.1	83.4	82.6	84.5	84.4
Ours Model	91.2	90.5	89.3	92.0	90.7

Figure 11 shows the AP (Average Precision) values of different object detection models on the dataset provided by the power supply company in Jilin City, measuring their detection performance across different categories (trash, ribbons, sunshades, and bird nests). It can be observed that there are significant differences in AP values across R-CNN, SPP, YOLOv3-tiny, YOLOv5m, and YOLOv8s models.

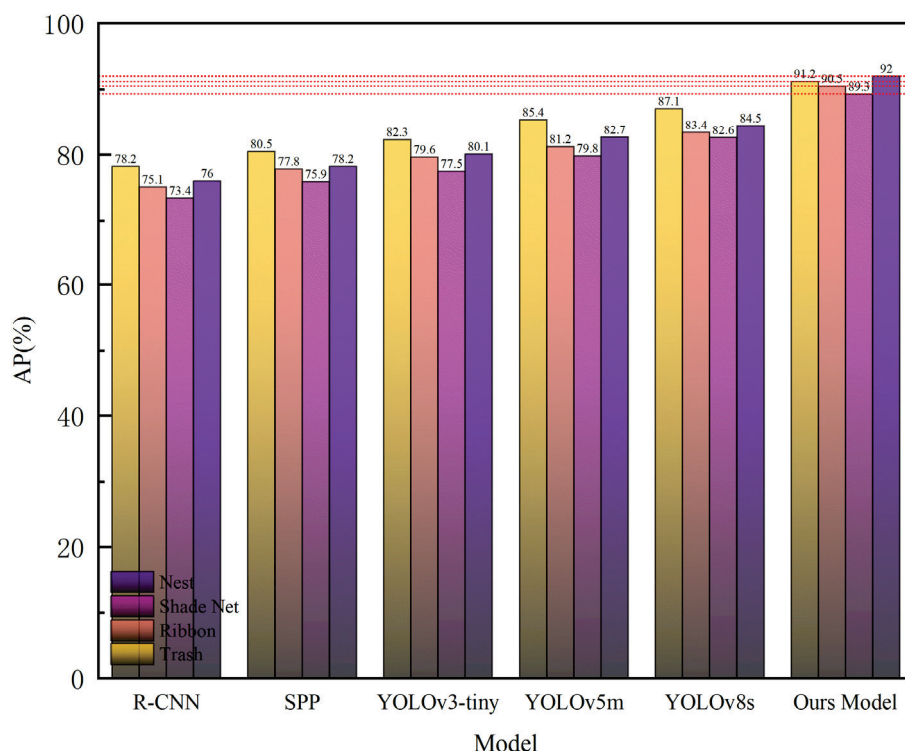


Figure 11. AP performance of different models on the dataset.

R-CNN achieved AP values of 78.2%, 75.1%, 73.4%, and 76% for trash, ribbons, sunshades, and bird nests, respectively. SPP showed a slight improvement, with AP values of 80.5%, 77.8%, 75.9%, and 78.2% for the same categories. YOLOv3-tiny further increased performance, with AP values of 82.3%, 79.6%, 77.5%, and 80.1%. YOLOv5m performed even better, with AP values of 85.4%, 81.2%, 79.8%, and 82.7%. YOLOv8s also demonstrated excellent AP values, achieving 87.1%, 83.4%, 82.6%, and 84.5%.

Notably, our model (AFPN + YOLOv8) achieved significantly higher AP values across all categories, with 91.2% for trash, 90.5% for ribbons, 89.3% for sunshades, and 92% for bird nests, with an overall average AP value of 90.7%. This indicates that the AFPN + YOLOv8 model significantly outperforms other models in object detection performance on the dataset, especially in complex scenes and small object detection.

Figure 12 shows the mAP@0.5 (%) performance metric of different object detection models on the dataset. mAP@0.5 is a common metric for measuring the accuracy of object detection models, with higher values indicating better detection performance. From the figure, it can be seen that R-CNN has a relatively low mAP of 75.7%; SPP shows a slight improvement with an mAP of 78.1%; YOLOv3-tiny's mAP further increases to 79.9%; YOLOv5m achieves an mAP of 82.3%; YOLOv8s reaches an mAP of 84.4%. In comparison, the AFPN + YOLOv8 (our model) significantly outperforms other models with an mAP of 90.7%.

This substantial performance improvement demonstrates that the combination of AFPN with YOLOv8 provides higher precision in object detection tasks, particularly in handling complex backgrounds and small object detection. This shows that AFPN + YOLOv8 not only has theoretical advantages but also significantly enhances detection performance in practical applications. Therefore, AFPN + YOLOv8 is an efficient and highly accurate object detection solution with broad application prospects.

Table 6 and Figure 13 shows the performance comparison of different object detection models on a dataset of over 1800 images, including R-CNN, SPP, YOLOv3-tiny, YOLOv5m, YOLOv8s, and AFPN + YOLOv8 (our model). The evaluated metrics include Precision, Recall, F1-score, mAP@0.5, GFLOPs (Giga Floating Point Operations per Second), and Inference Speed.

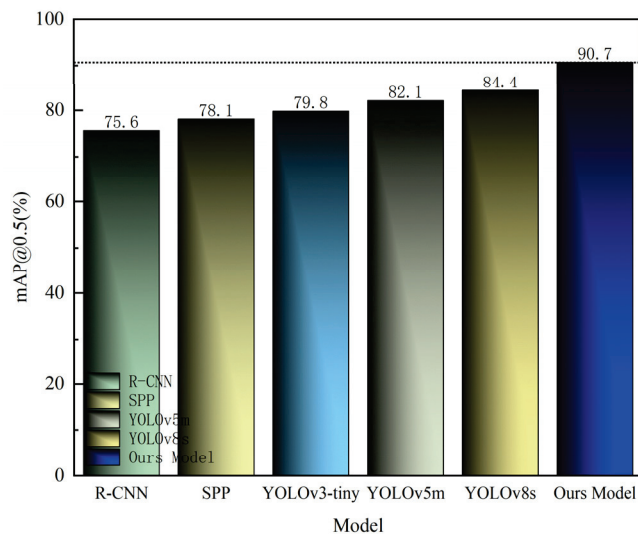


Figure 12. mAP performance of different models on the dataset.

Table 6. Comparison of evaluation metrics for different models.

<i>Model</i>	<i>Precision%</i>	<i>Recall%</i>	<i>F1-Score%</i>	<i>mAP@0.5</i>	<i>FPS</i>	<i>Inference Speed</i>
R-CNN	77.5	75.3	76.4	74.5	180	200
SPP	79.8	77.6	78.7	77.2	90	151.3
YOLOv3-tiny	81.0	78.9	79.9	78.5	17.9	30.8
YOLOv5m	83.2	81.0	82.1	81.0	36	25.6
YOLOv8s	85.0	83.2	84.1	83.5	27	16.5
Ours Model	88.5	87.0	87.7	89.3	14.3	11.2

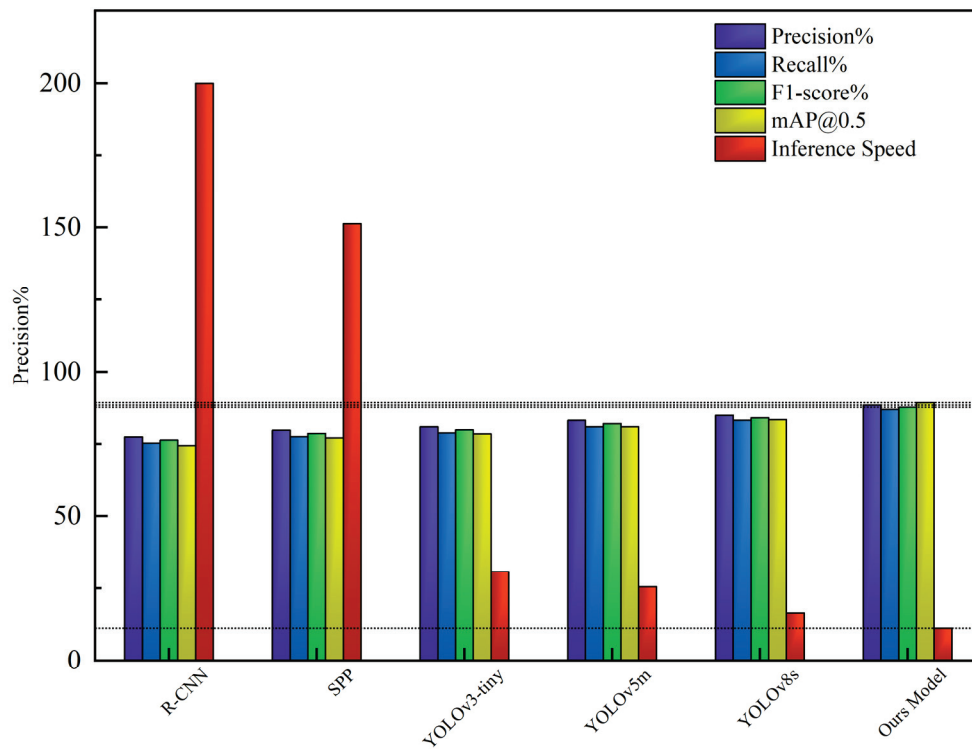


Figure 13. Evaluation metrics of different models.

R-CNN has a Precision of 77.5%, Recall of 75.3%, F1-score of 76.4%, mAP@0.5 of 74.5%, GFLOPs of 180, and Inference Speed of 200 ms, indicating high computational complexity and slower inference speed. SPP shows improved precision at 79.8%, with Recall of 77.6%, F1-score of 78.7%, mAP@0.5 of 77.2%, GFLOPs of 90, and Inference Speed of 151.3 ms, with improved precision but still relatively slow speed. YOLOv3-tiny achieves a Precision of 81.0%, Recall of 78.9%, F1-score of 79.9%, mAP@0.5 of 78.5%, GFLOPs of 17.9, and Inference Speed of 30.8 ms, maintaining low computational complexity and fast inference speed but slightly lower precision. YOLOv5m achieves a balance between precision and speed with a Precision of 83.2%, Recall of 81.0%, F1-score of 82.1%, mAP@0.5 of 81.0%, GFLOPs of 36, and Inference Speed of 25.6 ms. YOLOv8s shows good performance in both precision and speed, with a Precision of 85.0%, Recall of 83.2%, F1-score of 84.1%, mAP@0.5 of 83.5%, GFLOPs of 27, and Inference Speed of 16.5 ms.

AFPN + YOLOv8 (our model) demonstrates outstanding detection performance with a Precision of 88.5%, Recall of 87.0%, F1-score of 87.7%, mAP@0.5 of 89.3%, GFLOPs of 14.3, and Inference Speed of 11.2 ms. It shows excellent detection capabilities, low computational complexity, and very fast inference speed. Overall, AFPN + YOLOv8 outperforms other models in all performance metrics, particularly in Precision, Recall, F1-score, and mAP@0.5, while maintaining low GFLOPs and fast inference speed, making it suitable for applications requiring high accuracy and real-time performance.

4.4.3. Effect of Improved Loss Functions on YOLOv8

As shown in Table 7 and Figures 14–16, we can analyze the impact of different loss functions on model performance during training. It can be seen that the loss values for all loss functions drop rapidly at the beginning of training and gradually stabilize after about 50 epochs. Notably, the loss values for Focal SIoU and Focal CIoU are significantly lower than those of other loss functions throughout the training process, indicating superior performance in both convergence speed and final loss value. The loss value for WIoU is relatively high during training, indicating slower convergence and less effectiveness compared to other loss functions. The performance of CIoU and EIoU is intermediate and stable but not as excellent as the Focal loss functions.

Table 7. Comparison of loss values for different loss functions.

<i>LossFunction Type</i>	<i>Precision%</i>	<i>Recall%</i>	<i>mAP@0.5</i>	<i>Loss</i>
WIoU	88.1	78.2	85.9	0.608
CIoU(Initial)	89.7	79.3	86.7	0.412
EIoU	87.6	80.5	86.3	0.317
Focal CIoU	91.5	80.9	88.2	0.425
Focal SIoU	94.4	82.0	89.9	0.218

Figures 15 and 16 show the mAP@0.5 (mean Average Precision at an IoU threshold of 0.5) trends for different loss functions during training. It can be seen that the mAP@0.5 for all loss functions rises rapidly at the beginning of training and stabilizes after about 100 epochs. Overall, the mAP@0.5 for Focal SIoU and Focal CIoU is slightly higher than other loss functions in the later stages of training, which corresponds with the loss trends in Figure 17, further indicating that these two loss functions perform better in terms of accuracy and robustness. The mAP@0.5 for WIoU fluctuates significantly during training, performing slightly worse than other loss functions. The mAP@0.5 for EIoU and CIoU is similar, even and stable, showing moderate performance.

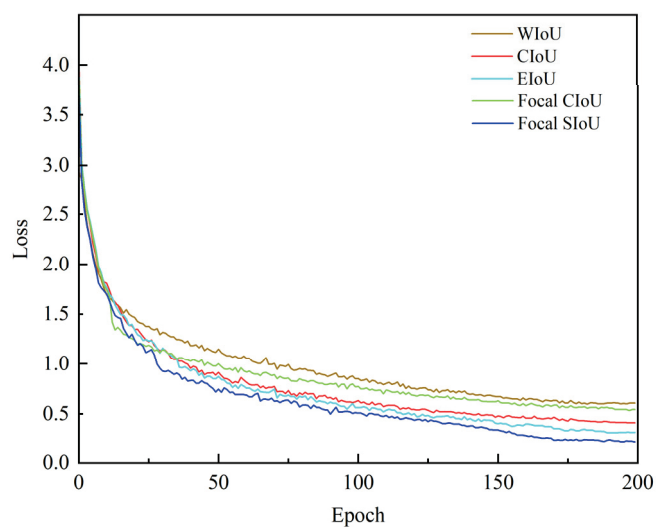


Figure 14. Loss values for different loss functions.

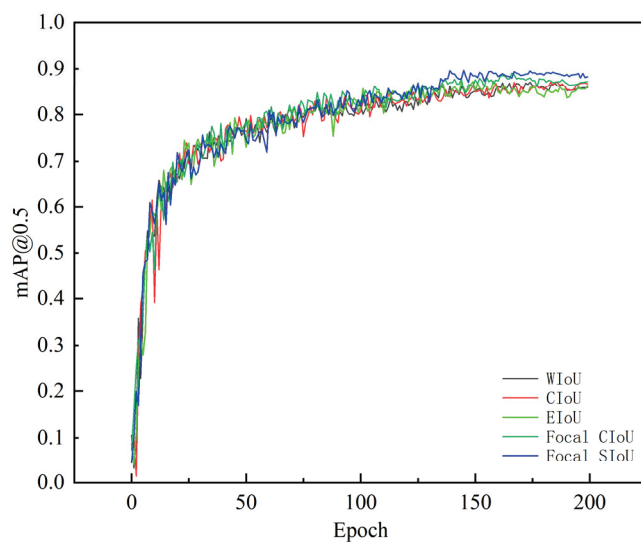


Figure 15. mAP for different loss functions.

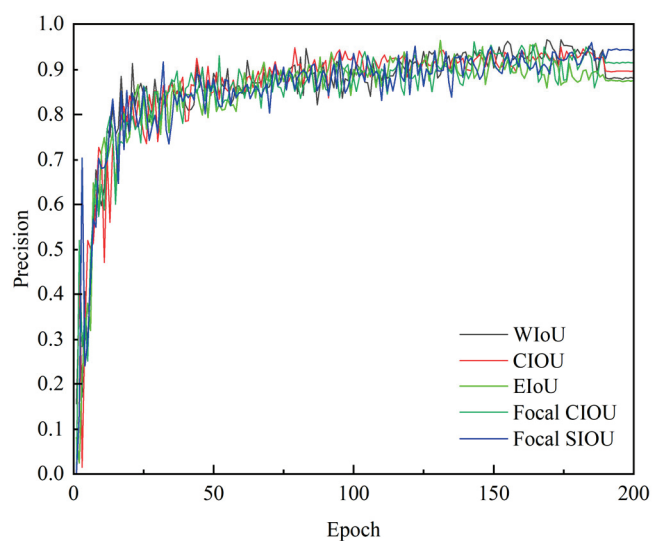


Figure 16. Precision for different loss functions.



(a) Faster R-CNN model



(b) YOLOv3-tiny model



(c) YOLOv3 model



(d) YOLOv5 model



(e) YOLOv7 model

Figure 17. *Cont.*



(f) YOLOv8 model



(g) Ours model

Figure 17. Compared to other advanced algorithms, our model has a high accuracy in detecting foreign objects.

Through the above analysis, we can see the importance of loss functions in model training. Different loss functions not only affect the convergence speed and final loss value of the model but also significantly impact the model's accuracy. Focal SIoU and Focal CIoU performed excellently in this experiment, outperforming other loss functions in both loss reduction speed and final mAP@0.5 performance. This indicates that Focal loss functions are more efficient and robust in handling object detection tasks. In contrast, WIoU is unsuitable for such tasks due to its unstable performance. EIoU and CIoU performed moderately; although not as superior as the Focal loss functions, their stable performance suggests they still have certain advantages in some application scenarios.

In summary, these results highlight the importance of selecting an appropriate loss function when training object detection models. By choosing the right loss function, model training efficiency and detection performance can be significantly improved, resulting in better outcomes in practical applications.

4.5. Ablation Study

In this section, we conduct an ablation study to analyze the impact of different model components on YOLOv8. An ablation study is a common method to evaluate the specific contributions of various parts of a model by gradually removing or replacing them. Table 8 shows the ablation experiment configurations for our model under seven different experimental setups, including the three main components: Swin Transformer, AFPN, and Focal SIoU. The specific setup of each experiment and the corresponding GFLOPs and mAP@0.5 (%) results are shown in the table. Specifically, Experiment 1 uses only the basic YOLOv8 model as the control group, Experiment 2 adds Swin Transformer to evaluate its performance improvement, Experiment 3 adds the AFPN module, and Experiment 4 introduces the Focal SIoU loss function. In Experiments 5 to 7, we combine these components to observe their synergistic effects on model performance. By comparing these experimental results, we can discuss in detail the impact of each component on the accuracy and computational complexity of the YOLOv8 model, thus guiding us in more effective model design and optimization in practical applications. The ablation experiments are shown in Table 8.

Table 8. Ablation study.

Experiment Number	Model			mAP@0.5 (%)
	SwinTransformer	AFPN	Focal SIoU	
1	✓			82.3
2		✓		81.7
3			✓	81.5
4	✓	✓		83.1
5	✓		✓	84.6
6		✓	✓	86.5
7	✓	✓	✓	89.7

We analyzed the impact of different model components on YOLOv8 performance through ablation experiments. The table shows the specific configurations and their corresponding mAP@0.5 (%) values for the seven experiments. Experiment 1 uses only the basic YOLOv8 model, achieving an mAP@0.5 of 82.3% as the baseline control group. Experiment 2 adds the Swin Transformer, increasing the mAP@0.5 to 81.7%, indicating a significant contribution. Experiment 3 incorporates the AFPN module, resulting in an mAP@0.5 of 81.5%, showing a certain improvement. Experiment 4 introduces the Focal SIoU loss function, raising the mAP@0.5 to 83.1%, confirming its effectiveness. Experiment 5 combines the Swin Transformer and AFPN, boosting the mAP@0.5 to 84.6%. Experiment 6 combines the Swin Transformer and Focal SIoU, achieving an mAP@0.5 of 86.5%. Experiment 7 combines all components, reaching the highest mAP@0.5 of 89.7%, demonstrating the best performance.

4.6. Comparison of the Combined Model with Other Advanced Algorithms

In this section, we will compare our combined model (Swin Transformer + AFPN + Focal SIoU) with other advanced algorithms in terms of precision, recall, mAP, and other performance metrics.

Based on Table 9 and Figure 17, we can observe significant differences among various object detection models in terms of precision, recall, mAP, computational complexity (FPS), and the number of parameters (Params). The Faster R-CNN model has the highest number of parameters, at 129 M, but its precision and recall are relatively low, at 61.3% and 42.5%, respectively. In contrast, YOLOv3-tiny, with only 14.5 M parameters, also shows lower precision and recall, at 71.8% and 69.2%, respectively. The YOLOv3 and YOLOv5 models demonstrate improved performance, especially YOLOv5, which achieves 84.0% precision and 84.0% recall with 24.69 M parameters, indicating a good balance. The YOLOv7 and YOLOv8 models further enhance performance, particularly YOLOv8, which reaches 87.2% precision and 83.9% recall while maintaining a parameter count of 28.4 M, showcasing better optimization.

Table 9. Performance comparison of different models and the combined model.

Model	Model				
	Precision (%)	Recall (%)	mAP@0.5 (%)	FPS	Params (M)
Faster R-CNN	61.3	42.5	74.3	38.7	129
YOLOv3-tiny	71.8	69.2	76.5	46.1	34.3
YOLOv3	82.8	75.5	83.8	45.5	61.9
YOLOv5	84.8	80.4	84.7	109.8	24.69
YOLOv7	86.2	80.45	87.1	117.5	28.67
YOLOv8	87.2	83.9	88.2	150.7	28.40
Ours Model	90.8	89.1	89.7	181.4	28.88

Our combined model performs exceptionally well across all metrics. It achieves a precision of 90.8%, a recall of 89.1%, an mAP of 89.7%, FPS of 181.4, and a parameter count of 28.88 M. Although its computational complexity is higher, its advantages in precision and recall make it highly valuable in practical applications. The image comparisons show that “Our Model” demonstrates higher confidence in detecting various objects (such as kites, nests, and balloons), validating its superior performance in real-world detection tasks.

Overall, our combined model ensures high precision and recall while maintaining a moderate number of parameters and computational complexity, making it an efficient and practical object detection model. These excellent performance metrics make it highly promising for practical applications, especially in scenarios requiring high precision and real-time performance.

5. Conclusions

In this study, we proposed an enhanced YOLOv8-based model for detecting foreign objects on power transmission lines, incorporating the Swin Transformer, AFPN (Asymptotic Feature Pyramid Network), and a novel loss function, Focal SIOU. This work aims to address the challenges of accurate and real-time detection of foreign objects such as bird nests, kites, and balloons, which pose significant risks to power transmission line stability and safety.

The integration of the Swin Transformer into the YOLOv8 backbone network significantly improves feature extraction capabilities. AFPN enhances the multi-scale feature fusion process, effectively integrating information from different levels and improving detection accuracy, especially for small and occluded objects. The introduction of the Focal SIOU loss function optimizes the model’s training process, enhancing its ability to handle hard-to-classify samples and uncertain predictions.

Our experiments demonstrate the superiority of the enhanced YOLOv8 model over other state-of-the-art object detection algorithms. The enhanced YOLOv8 achieves higher average precision (AP) across various categories, such as 91.2% for trash and 92% for bird nests, outperforming models like Faster R-CNN and SSD. The Focal SIOU loss function leads to notable improvements in precision and recall rates.

Despite higher computational complexity (181.4 FPS), our model maintains a moderate number of parameters (28.88 M) and an inference speed of 11.2 ms, making it suitable for real-time applications. The proposed enhancements make YOLOv8 a highly effective solution for detecting foreign objects on power transmission lines, significantly contributing to the safety and reliability of power supply systems.

Author Contributions: Methodology, H.S.; Software, Q.S.; Validation, H.K.; Writing—review & editing, Z.D. and X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Scientific and Technological Planning Project of Jilin Province (YDZJ202201ZYTS642).

Data Availability Statement: Due to security and confidentiality concerns, the data generated and analyzed during the study on foreign object intrusion in power transmission lines are not available for public disclosure. The data involve sensitive information related to critical infrastructure, and making them public could compromise the safety and stability of the power grid. Therefore, these data will not be shared or made publicly available.

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

References

1. Zhang, D.; Zhang, Z.; Zhao, N.; Wang, Z. A Lightweight Modified YOLOv5 Network Using a Swin Transformer for Transmission-Line Foreign Object Detection. *Electronics* **2023**, *12*, 3904. [CrossRef]
2. Yu, Y.; Qiu, Z.; Liao, H.; Wei, Z.; Zhu, X.; Zhou, Z. A Method Based on Multi-Network Feature Fusion and Random Forest for Foreign Objects Detection on Transmission Lines. *Appl. Sci.* **2022**, *12*, 4982. [CrossRef]
3. Wu, M.; Guo, L.; Chen, R.; Du, W.; Wang, J.; Liu, M.; Kong, X.; Tang, J. Improved YOLOX Foreign Object Detection Algorithm for Transmission Lines. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5835693. [CrossRef]

4. Tasnim, S.; Qi, W. Progress in Object Detection: An In-Depth Analysis of Methods and Use Cases. *Eur. J. Electr. Eng. Comput. Sci.* **2023**, *7*, 39–45. [CrossRef]
5. Ren, J.; Wang, J. Overview of Object Detection Algorithms Using Convolutional Neural Networks. *J. Comput. Commun.* **2022**, *10*, 115–132.
6. Mishra, D. Deep Learning based Object Detection Methods: A Review. *J. Comput. Commun.* **2022**, *10*, 115–132.
7. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
8. Shi, Y.; Jia, Y.; Zhang, X. FocusDet: An efficient object detector for small object. *Sci. Rep.* **2024**, *14*, 10697. [CrossRef] [PubMed]
9. Wang, X.; Wang, A.; Yi, J.; Song, Y.; Chehri, A. Small Object Detection Based on Deep Learning for Remote Sensing: A Comprehensive Review. *Remote Sens.* **2023**, *15*, 3265. [CrossRef]
10. Min, L.; Fan, Z.; Lv, Q.; Reda, M.; Shen, L.; Wang, B. YOLO-DCTI: Small Object Detection in Remote Sensing Base on Contextual Transformer Enhancement. *Remote Sens.* **2023**, *15*, 3970. [CrossRef]
11. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
12. Zhang, Y.; Shen, T. Small Object Detection with Multiple Receptive Fields. IOP conference series. *Earth Environ. Sci.* **2020**, *440*, 32093.
13. Cheng, G.; Yuan, X.; Yao, X.; Yan, K.; Zeng, Q.; Xie, X.; Han, J. Towards Large-Scale Small Object Detection: Survey and Benchmarks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 13467–13488. [CrossRef] [PubMed]
14. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. 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]
15. Flores-Calero, M.; Astudillo, C.A.; Guevara, D.; Maza, J.; Lita, B.S.; Defaz, B.; Ante, J.S.; Zabala-Blanco, D.; Moreno, J.M.A. Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review. *Mathematics* **2024**, *12*, 297. [CrossRef]
16. Kumawat, D.; Abhayankar, D.; Tanwani, S. Exploring Object Detection Algorithms and implementation of YOLOv7 and YOLOv8 based model for weapon detection. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *12*, 877–886.
17. Shao, Y.; Zhang, R.; Lv, C.; Luo, Z.; Che, M. TL-YOLO: Foreign-Object Detection on Power Transmission Line Based on Improved Yolov8. *Electronics* **2024**, *13*, 1543. [CrossRef]
18. Tang, J.; Li, B. Power System to Prevent External Damage Detection Method. *Acad. J. Sci. Technol.* **2024**, *9*, 189–193. [CrossRef]
19. Shi, Q.; Deng, K. Privacy-Preserving Detection Method for Transmission Line Based on Edge Collaboration. In Proceedings of the 2023 IEEE 22nd International Conference, Exeter, UK, 1–3 November 2023.
20. Li, M.; Ding, L. DF-YOLO: Highly Accurate Transmission Line Foreign Object Detection Algorithm. *IEEE Access* **2023**, *11*, 108398–108406. [CrossRef]
21. Liu, Y.; Gao, M.; Zong, H.; Wang, X.; Li, J. Real-Time Object Detection for the Running Train Based on the Improved YOLO V4 Neural Network. *J. Adv. Transp.* **2022**, *2022*, 4377953. [CrossRef]
22. Nayak, R.; Behera, M.M.; Pati, U.C.; Das, S.K. Video-based Real-time Intrusion Detection System using Deep-Learning for Smart City Applications. In Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019.
23. Chen, W.; Li, Y.; Li, C. A Visual Detection Method for Foreign Objects in Power Lines Based on Mask R-CNN. *Int. J. Ambient. Comput. Intell.* **2020**, *11*, 34–47. [CrossRef]
24. Sevi, M.; Aydin, İ. Detection of Foreign Objects Around the Railway Line with YOLOv8. *Comput. Sci.* **2023**, *25*, 19–23. [CrossRef]
25. Shan, H.; Song, Y.; Wang, H.; Chen, Y. Research on Efficient Detection Method of Foreign Objects on Transmission Lines Based on Improved YOLOv4 Network. *J. Phys. Conf. Ser.* **2022**, *2404*, 012040. [CrossRef]
26. Li, S.; Xie, J.; Zhou, F.; Liu, W.; Li, H. Foreign Object Intrusion Detection on Metro Track Using Commodity WiFi Devices with the Fast Phase Calibration Algorithm. *Sensors* **2020**, *20*, 3446. [CrossRef] [PubMed]
27. Wang, Z.; Yuan, G.; Zhou, H.; Ma, Y.; Ma, Y. Foreign-Object Detection in High-Voltage Transmission Line Based on Improved YOLOv8m. *Appl. Sci.* **2023**, *13*, 12775. [CrossRef]
28. Zhao, L.; Liu, C.A.; Qu, H. Transmission Line Object Detection Method Based on Contextual Information Enhancement and Joint Heterogeneous Representation. *Sensors* **2022**, *22*, 6855. [CrossRef] [PubMed]
29. Chien, C.T.; Ju, R.Y.; Chou, K.Y.; Lin, C.S.; Chiang, J.S. YOLOv8-AM: YOLOv8 with Attention Mechanisms for Pediatric Wrist Fracture Detection. 2024, Cornell University Library. *arXiv* **2024**, arXiv:2402.09329.
30. Liu, Q.; Ye, H.; Wang, S.; Xu, Z. YOLOv8-CB: Dense Pedestrian Detection Algorithm Based on In-Vehicle Camera. *Electronics* **2024**, *13*, 236. [CrossRef]
31. Ren, R.; Sun, H.; Zhang, S.; Wang, N.; Lu, X.; Jing, J.; Xin, M.; Cui, T. Intelligent Detection of Lightweight “Yuluxiang” Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy* **2023**, *13*, 2418. [CrossRef]
32. Ju, R.Y.; Cai, W. Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. *Sci. Rep.* **2023**, *13*, 20077. [CrossRef]
33. Baek, J.; Chung, K. Swin Transformer-Based Object Detection Model Using Explainable Meta-Learning Mining. *Appl. Sci.* **2023**, *13*, 3213. [CrossRef]

34. Huo, D.; Kastner, M.A.; Liu, T.; Kawanishi, Y.; Hirayama, T.; Komamizu, T.; Ide, I. Small Object Detection for Birds with Swin Transformer. In Proceedings of the 2023 18th International Conference on Machine Vision and Applications (MVA), Hamamatsu, Japan, 23–25 July 2023.
35. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 10012–10022.
36. Peng, Y.; Ren, J.; Wang, J.; Shi, M. Spectral-Swin Transformer with Spatial Feature Extraction Enhancement for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 2696. [CrossRef]
37. 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.
38. Schuster, R.; Battaway, R.; Wasenmüller, O.; Stricker, D. ResFPN: Residual Skip Connections in Multi-Resolution Feature Pyramid Networks for Accurate Dense Pixel Matching. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021.
39. Dimitrakopoulos, P.; Sfikas, G.; Nikou, C. Nuclei Detection Using Residual Attention Feature Pyramid Networks. In Proceedings of the 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 28–30 October 2019.
40. Wang, B.; Ji, R.; Zhang, L.; Wu, Y. Bridging Multi-Scale Context-Aware Representation for Object Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 2317–2329. [CrossRef]
41. Zhang, D.; Zhang, H.; Tang, J.; Wang, M.; Hua, X.; Sun, Q. Feature Pyramid Transformer. In *Computer Vision—ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020.
42. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic Feature Pyramid Network for Object Detection. In Proceedings of the 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023.
43. Hu, D.; Yu, M.; Wu, X.; Hu, J.; Sheng, Y.; Jiang, Y.; Huang, C.; Zheng, Y. DGW-YOLOv8: A small insulator target detection algorithm based on deformable attention backbone and WIoU loss function. *IET Image Process.* **2024**, *18*, 1096–1108. [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.

Article

Knowledge Distillation in Image Classification: The Impact of Datasets

Ange Gabriel Belinga^{1,*}, Cédric Stéphane Tekouabou Koumetio^{2,3,*}, Mohamed El Haziti^{1,4}
and Mohammed El Hassouni⁵

¹ Laboratory of Research in Computer Science and Telecommunications (LRIT), Faculty of Sciences in Rabat, Mohammed V University in Rabat, Rabat 10000, Morocco

² Laboratory in Computer Science and Educational Technologies (LITE), Higher Teacher Training College (HTTC), University of Yaoundé 1, Yaounde P.O. Box 47, Cameroon

³ Department of Computer Science and Educational Technologies (DITE), Higher Teacher Training College (HTTC), University of Yaoundé 1, Yaounde P.O. Box 47, Cameroon

⁴ High School of Technology, Mohammed V University in Rabat, Sale 11000, Morocco; elhazitim@gmail.com

⁵ FLSH, Mohammed V University in Rabat, 3 Av. Ibn Batouta, Rabat 10090, Morocco; mohamed.elhassouni@flsh.um5.ac.ma

* Correspondence: agbelinga@gmail.com (A.G.B.); ctekouaboukoumetio@gmail.com (C.S.T.K.); Tel.: +212-771-515-963 or +237-697-64-69-78 (C.S.T.K.)

Abstract: As the demand for efficient and lightweight models in image classification grows, knowledge distillation has emerged as a promising technique to transfer expertise from complex teacher models to simpler student models. However, the efficacy of knowledge distillation is intricately linked to the choice of datasets used during training. Datasets are pivotal in shaping a model's learning process, influencing its ability to generalize and discriminate between diverse patterns. While considerable research has independently explored knowledge distillation and image classification, a comprehensive understanding of how different datasets impact knowledge distillation remains a critical gap. This study systematically investigates the impact of diverse datasets on knowledge distillation in image classification. By varying dataset characteristics such as size, domain specificity, and inherent biases, we aim to unravel the nuanced relationship between datasets and the efficacy of knowledge transfer. Our experiments employ a range of datasets to comprehensively explore their impact on the performance gains achieved through knowledge distillation. This study contributes valuable guidance for researchers and practitioners seeking to optimize image classification models through knowledge distillation. By elucidating the intricate interplay between dataset characteristics and knowledge distillation outcomes, our findings empower the community to make informed decisions when selecting datasets, ultimately advancing the field toward more robust and efficient model development.

Keywords: knowledge distillation; dataset selection; transfer learning; knowledge transfer

1. Introduction

In the ever-evolving landscape of computer vision, image classification is a fundamental and challenging task with many applications [1–4]. In the recent literature, machine learning (ML) models based on deep neural networks (DNNs) have proven to be the most effective for computer vision, particularly image analysis [5–7]. To achieve this efficiency, several DNN architectures have been proposed in the literature, with different processes including knowledge distillation [8–10]. Knowledge distillation in deep neural networks is a crucial process in the ML field [11]. As the demand for more efficient and lightweight models grows, the concept of knowledge distillation (KD) has emerged as a promising avenue to transfer knowledge from complex, high-capacity models (teachers) to simpler, more deployable counterparts (students) [8,12]. This transfer of knowledge from the teacher to the student through a training paradigm typically involves the following steps.

1. Teacher model training: The first step is to train a large and complex model (the teacher) on a given dataset to achieve high accuracy.
2. Generation of soft targets: The trained teacher model is then used to make predictions on the training data, producing probability distributions (soft targets) over possible classes. These soft targets contain more information than the hard targets (i.e., the actual labels), as they reflect the relative confidence of the teacher model in its predictions. The soft targets can be obtained using a softmax function

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \quad (1)$$

where q_i is the output probability for class i , z_i is the logit for class i , and T is the temperature parameter.

3. Student model training: The smaller student model is trained using a combination of the original true labels and the soft targets generated by the teacher model. The loss function typically includes a component for standard classification loss and another component for distillation loss, which measures the difference between the student and teacher probability distributions. The Kullback–Leibler (KL) function is usually used for distillation loss. The KL formula is defined as

$$D_{KL}(P \parallel Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \quad (2)$$

where P and Q are probability distributions defined on the same sample space X . The final loss formula is defined as

$$L_{total} = \alpha L_{classification} + (1 - \alpha) L_{distillation} \quad (3)$$

where $L_{classification} = \sum_i y_i \log(p_i)$ and $L_{distillation} = KL(q_{teacher}^T \parallel q_{student}^T)$.

Indeed, this approach makes it possible to compress and generalize the information learned by complex deep neural networks, facilitating their deployment on resource-limited devices [8,13]. This process of KD not only facilitates model compression but also enhances the generalization capabilities of the student model [10]. The success of KD is inherently tied to the quality and diversity of the datasets used during the training step, as well as the large applications of the KD learning-based processes [1,12,14–19].

The effectiveness of KD in DNN could depend largely on the complexity (quality and quantity, etc.) of the data used. Thus, datasets play a pivotal role in shaping the learning process, influencing the model's ability to discern patterns and generalize to unseen features [15–17]. While extensive research has been conducted on KD and image classification independently, a comprehensive understanding of how various datasets impact the effectiveness of KD remains an open and critical area of investigation. However, although many studies have been published on this method, few have explored in depth how the characteristics and properties of the data would influence this knowledge transfer process. This research gap raises a crucial question: How do data characteristics, such as complexity, diversity, and distribution, impact the efficiency of KD in a deep neural network? Answering this question will enable us to better understand the challenges and opportunities for KD applications related to the use of different data sources, paving the way for more efficient and robust techniques for transfer learning in deep neural networks.

This study seeks to address this gap by systematically examining the impact of different datasets on KD in image classification. As datasets vary in terms of size, domain specificity, and inherent biases, their influence on the transfer of knowledge from teacher to student models warrants meticulous exploration. Through a series of experiments, we aim to unravel the intricate relationship between dataset characteristics and the performance gains achieved through knowledge distillation. In the subsequent sections, we delve into the relevant literature, providing insights into the existing landscape of KD and its appli-

cation in image classification. Following this, we elucidate our methodology, detailing the datasets chosen for experimentation, model architectures, and the KD process. The results and their implications are then discussed, shedding light on the nuanced impact of datasets. Ultimately, this study aims to contribute valuable insights for researchers and practitioners navigating the intersection of knowledge distillation and image classification, offering guidance on optimizing model performance through judicious dataset selection.

Following on from the remainder of this work, Section 2 will discuss previous work on knowledge distillation in deep neural networks. Then, Sections 3 and 4 will describe the proposed research approach and analyze the obtained results, respectively. Finally, Sections 5 and 6 will discuss the results obtained and conclude this work.

2. Related Work

Knowledge distillation (KD) has been widely studied in the literature, and several notable works have contributed to the understanding and development of this technique [20–25]. Since its introduction by Hinton et al. [8], this approach has attracted growing interest in the machine learning research community. Table 1 presents some recent knowledge distillation work in the field of image classification. This table mainly presents the different databases, the architecture of the teacher and student models and the main evaluation metric used to perform KD in image classification task.

2.1. Knowledge Distillation in the Literature

Several works have explored various aspects of knowledge distillation in deep neural networks [26], including teacher and student model architectures, regularization techniques, and optimization methods.

For example, Li et al. proposed a transferred attention method to improve the performance of convolutional neural networks [27], while Yazdanbakhsh et al. studied the application of knowledge distillation in specific domains such as healthcare [19]. However, despite these significant advances, little attention has been paid to the impact of data on this knowledge transfer process.

Table 1. Summary of recent literature on knowledge distillation in image classification. EM = evaluation metric.

-	About Data				Methods		EM
Ref	Year	Type	Dataset	Type	Teacher	Student	Acc
LightCyan [8]	2015	article	JFT,MNIST	Images	DNN	DNN	✓
[28]	2019	article	PPMI, Willow, UIUC-Sport	Images	AlexNet	AlexNet	✓
LightCyan [29]	2021	article		Images	DNN	DNN	✓
[30]	2017	Conf	CIFAR100	Images	VGG13, ResNet32x4	VGG13, ResNet32x4	✓
LightCyan [31]	2018	Conf	CIFAR100	Images	ResNet34	VGG9	✓
[32]	2020	Conf	CIFAR10, CIFAR100	Images	ResNet26	ResNet8&14	✓
LightCyan [20]	2014	article	CIFAR10, CIFAR100, SVHN, MNIST, AFLW	Images		FiNet	✓
[33]	2021	article	CIFAR100	Images	ResNet20	ResNet8	✓
LightCyan [34]	2024	article	CIFAR100, ImageNet	Images	ResNet152	ResNet18	✓
[35]	2023	article	CIFAR100	Images	ResNet50& ResNet34	ResNet18	✓
LightCyan [36]	2023	article	CIFAR100	Images	ResNet18	ResNet18	✓

The authors demonstrated the effectiveness of the distillation on various tasks and highlighted its potential for model compression. The FitNets paper [20] proposed a specific form of knowledge distillation called FitNets, where a student network is guided not only by the output probabilities of a teacher network but also by intermediate representations

(or hints). This work aimed to improve the transfer of information in the training process. Ref. [27] introduces attention transfer as a form of knowledge distillation. It focuses on transferring attention maps from a teacher to a student network to improve the student's performance. Attention transfer has proven effective in enhancing the generalization capabilities of the student model. To address the limitations of traditional knowledge distillation, ref. [31] introduces Jacobian matching, a novel method that aims to transfer not only the output probabilities but also the derivatives of the teacher model's predictions. This approach provides a more comprehensive form of knowledge transfer. Ref. [30] explores the benefits of knowledge distillation beyond model compression. The authors show that the knowledge distillation process not only compresses models but also accelerates the optimization process, enabling faster convergence during training. Ref. [32] introduces the concept of a "teacher assistant" by proposing an extension to traditional knowledge distillation. The teacher assistant helps bridge the performance gap between the teacher and the student, leading to enhanced knowledge transfer.

2.2. Role of Datasets for Model Training by KD

The impact of datasets on model training has been a longstanding focus in machine learning research. Datasets serve as the foundation upon which models learn to recognize and classify patterns, making their composition and characteristics crucial determinants of model performance. Studies by refs. [37,38] emphasize the importance of diverse datasets in fostering robust image recognition systems, highlighting how exposure to a wide range of scenarios aids in generalization. In the context of image classification, biases present in datasets have been identified as potential challenges, leading to models that may not generalize well across different domains [37]. Addressing these biases and ensuring dataset diversity are pivotal considerations in the pursuit of building models that can perform reliably across various real-world scenarios.

2.3. Research Gap and Motivation

While the individual importance of KD and dataset characteristics in image classification has been adequately explored, a comprehensive examination of how different datasets impact the success of KD remains a notable gap in the literature. Synthesizing the existing literature, we recognize the intertwined nature of knowledge distillation and dataset influence on image classification models. Furthermore, the literature review confirms the preliminary observation that several works have studied knowledge distillation in neural networks [8,28,29,31,32,36]. However, the majority of these studies have used not only a single dataset (CIFAR10, CIFAR100, MNIST, ImageNet, etc.) [20,30–32,34] but also, more often than not, residual network architectures (ResNet) [30,32–35]. Moreover, knowledge acquisition is relative to the context, which is nothing other than the data, whereas the existing studies often focus on benchmark datasets without thoroughly investigating the nuances introduced by varying dataset characteristics.

This study aims to bridge this gap by systematically exploring the relationship between dataset properties and the efficacy of knowledge distillation. Successful knowledge transfer relies not only on the distillation techniques but also on the inherent properties of the datasets used during training. In the subsequent sections, we detail our methodology, experimentally addressing this critical gap and shedding light on how different datasets impact the performance of knowledge-distilled models.

3. Research Method

The research approach adopted in this paper aims to highlight the impact of data complexity on knowledge distillation in deep convolutional neural networks. To better illustrate this approach, we have represented its operating process on the diagram in Figure 1, which gives a better overview of the different steps of the followed method.

From this illustration, the first step in our approach is to select the databases most commonly used in the literature (see analysis in Section 2.2), which will enable us to carry

out our study, as detailed in the following Section 3.1. Once the databases have been selected, the next step is to choose the architectures of the teaching and learning neural networks with which to test our approach. Once the architectures have been chosen, this stage, which we detail in Section 3.2, ends with training the parent model and an instance of the student model from scratch on all the experiments' datasets. Then, the third stage of our experiment consists of simulating the trained student models through knowledge distillation according to two configurations, namely response-based distillation (RKD) and intermediate-based distillation (IKD), which we explain in Section 3.3. Finally, the fourth and last stage of our study consists of comparing the results and seeing the effect of different databases on knowledge distillation.

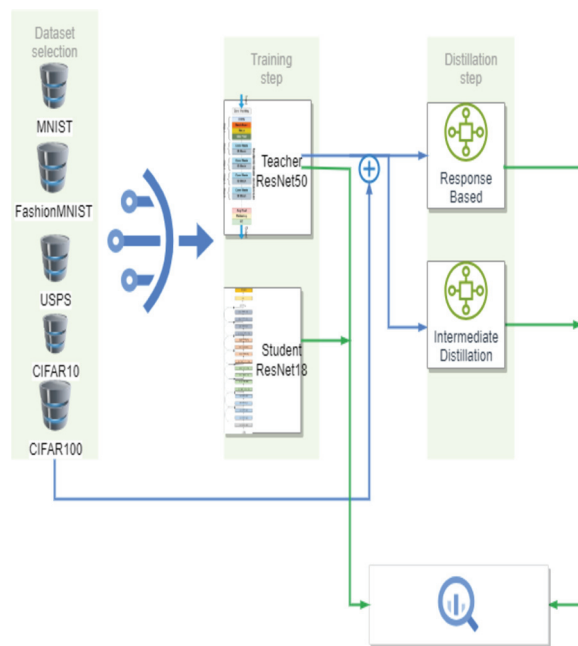


Figure 1. Flowchart of the proposed approach to highlight the impact of the dataset on knowledge distillation in DNN.

3.1. Datasets Selection

To comprehensively investigate the impact of datasets on knowledge distillation in image classification, a diverse set of datasets is curated. The selection criteria include considerations of size, domain specificity, and potential biases. Well-established benchmark datasets, such as CIFAR-10, CIFAR-100, and MNIST as shown in Table 1, form the core of our study, providing a foundation for cross-dataset comparisons.

Dataset Description and Complexity Classification

To highlight the impact of datasets on the distillation of knowledge learned by deep neural networks, we tested teacher and student network architectures on the most popular datasets in the scientific machine learning literature. For this purpose, we used 5 different data sets, including MNIST [6], FashionMNIST [7], UPS [11], CIFAR10 and CIFAR100 [5], which are summarized in Table 2 and described in turn in the rest of this section.

Each dataset was selected to represent different characteristics and complexities, ensuring a comprehensive evaluation of the distillation process. The classification of the level of data complexity in this article is based on a combined analysis of the dataset's characteristics (dimensionality, class diversity, data volume, variability and domain specificity) and the performances obtained in the literature [39,40]. Below are descriptions of the datasets mentioned in the literature review for knowledge distillation in image classification:

Table 2. The key statistics for each dataset.

Dataset	Image Sizes	Nb Classes	Nb-Images	Complexity Level
CIFAR-10	32×32	10	60,000	Moderate to high
CIFAR-100	32×32	100	60,000	High
USPS	16×16	10	9298	Moderate
MNIST	28×28	10	60,000	Low
Fashion MNIST	28×28	10	60,000	Low to moderate

- CIFAR-10 [5]: This dataset consists of 60,000 32×32 color images across ten different classes, each containing 6000 images. The classes include common objects like cars, dogs, and cats. The addition of color and more diverse objects increases the complexity compared to MNIST and USPS. Criteria: larger image size (32×32 pixels), three-channel color images, more diverse classes, and significant background variations.
- CIFAR-100 [5]: Similar to CIFAR-10, CIFAR-100 has 100 classes, with 600 images per class. It covers a broader range of object categories, making it more challenging. The increased number of classes and the finer distinctions between categories make it a more complex classification task compared to the previous datasets. Criteria: same image size (32×32 pixels) and color channels as CIFAR10, but a much larger number of classes (100), increasing variability and the challenge of classification.
- USPS [11] is a digit dataset automatically scanned from envelopes by the U.S. Postal Service containing a total of 9298 16×16 pixel grayscale samples; the images are centered and normalized and show a broad range of font styles. Similar to MNIST, USPS contains images of handwritten digits. It is slightly more challenging than MNIST but still relatively simple. Criteria: small image size (16×16 pixels), same number of classes (10 digits), and slight variations in style and noise compared to MNIST.
- MNIST [41] is a dataset with 28×28 grayscale images of handwritten digits. It consists of ten different classes and is often used for image classification tasks. The dataset is relatively simple and is often used as a beginner's dataset for image classification tasks. Criteria: small image size (28×28 pixels), a limited number of classes (10 digits), simple and uniform structure with minimal noise.
- Fashion MNIST [7] is a dataset with 28×28 grayscale images of fashion items, such as clothing and accessories. It consists of ten different classes and is often used as a replacement for the traditional MNIST dataset for image classification tasks. The dataset is more complex than MNIST as it requires the model to recognize various types of clothing items, adding a bit more complexity to the classification task. Criteria: same image size (28×28 pixels) as MNIST, but with 10 different classes of clothing, introducing more variability in shapes, and textures.

The levels of complexity of the datasets were determined according to several key criteria, which include the following:

- Dimensionality: the resolution and color channels of the images. higher resolution and multiple colour channels generally increase the complexity of the dataset, as they require more sophisticated models to capture detail.
- Class diversity: the number and variability of classes within the dataset. A larger number of classes with significant differences between them increases complexity because the model has to distinguish between a larger set of categories.
- Data volume: the size of the dataset in terms of the number of samples. Larger datasets can be more complex to manage and require more computing resources, but they also provide more information for robust model formation.
- Variability: the level of noise, background variation, and object diversity within the dataset. Datasets with high variability in object appearance, backgrounds, and noise levels are more difficult for models to learn and generalize.

- Domain specificity: the within-domain specificity and variability of the dataset (e.g., handwritten figures versus real-world objects). Datasets from domains with high intra-class variability and inter-class similarity are considered more complex due to the more subtle distinctions that need to be learned.

The complexity increases from MNIST and USPS to FashionMNIST, CIFAR-10, and finally CIFAR-100, with the latter being the most challenging among the mentioned datasets for an image classification task using the ResNet architecture.

3.2. Model Architecture Details

Our experimental setup involves employing state-of-the-art model architectures as both teacher and student networks. Convolutional neural networks (CNNs) [42,43] have demonstrated exceptional performance in image classification tasks [44,45], and we leverage ResNet [46] architectures for our experiments. Table 1 shows the frequency of use of ResNet in the literature. The teacher model, being more complex, serves as the knowledge source, while the student model is designed with fewer parameters to facilitate efficient deployment.

ResNet, introduced by ref. [46], has become a pivotal architecture in deep learning due to its ability to tackle the vanishing gradient problem through the innovative use of residual connections [14].

The key innovation of ResNet lies in the use of residual blocks (Figure 2), where each block contains a shortcut connection that bypasses one or more convolutional layers. This shortcut connection enables the network to learn residual mappings, making it easier to optimize deeper architectures. ResNet architectures come in various depths, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 [46], each with a different number of layers. The following Table 3 shows the characteristics of the models used in our experiment.

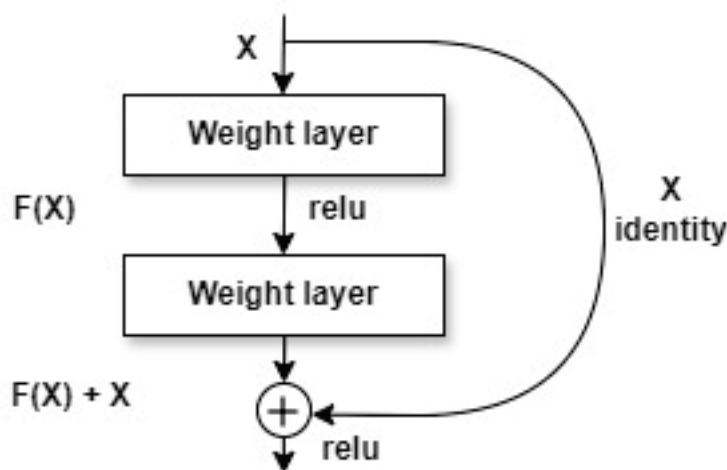


Figure 2. Residual blocks of ResNet architecture [46].

This table shows the key details of the ResNet-50 and ResNet-18 architectures [46] used in our experiments. The Bottleneck layers (ResNet-50) consist of three layers (1×1 convolution for channel reduction, 3×3 convolution, and 1×1 convolution for channel restoration), optimizing network efficiency and depth, and the Basic Unit layers (ResNet-18) consist of two 3×3 convolution layers, maintaining simplicity and reducing computational load.

Table 3. Details of the ResNet architectures used for the teacher and student models

Feature	ResNet50 (Teacher Model)	ResNet18 (Student Model)
Total layers	50	18
Initial Conv Layer	7×7 , 64, stride 2	3×3 Max Pool, stride 2
Initial Pooling Layer	7×7 , 64, stride 2	3×3 Max Pool, stride 2
Residual Block 1/Channels	3 Bottleneck Units/246	2 Basic Units/64
Residual Block 2/Channels	4 Bottleneck Units/512	2 Basic Units/128
Residual Block 3/Channels	6 Bottleneck Units/1024	2 Basic Units/256
Residual Block 4/Channels	3 Bottleneck Units/2048	2 Basic Units/512
Pooling Layer	Global Avg Pool	Global Avg Pool
Fully Connected Layer	1000-d FC Layer	1000-d FC Layer

3.3. Knowledge Distillation Processes

The knowledge distillation process involves transferring the knowledge from the teacher to the student model. We employ a combination of soft targets and intermediate representations during training. The soft targets, representing the teacher model's softened predictions, are integrated with traditional cross-entropy loss using the following formula.

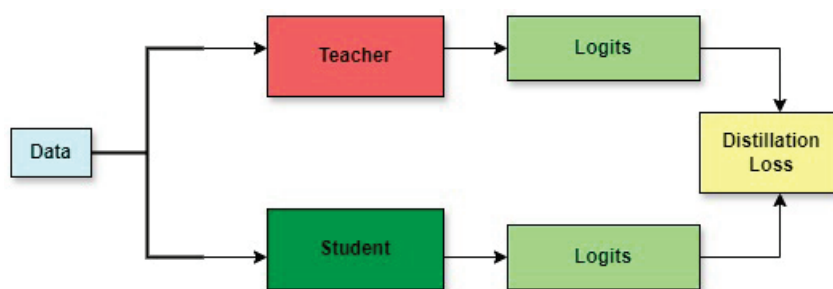
$$L_{KD} = (1 - \alpha)L_{CE}(y, P^{(s)}) + \alpha\tau^2 D_{KL}(P^{(t)} / \tau, P^{(s)} / \tau) \quad (4)$$

where $\alpha \in (0, 1)$ is the balance factor between the two loss terms; L_{CE} is the cross-entropy loss; y is the one-hot label; $P^{(t)}$ is the teacher output; $P^{(s)}$ is the student output; D_{KL} is the KL divergence [47]; and τ is a temperature [8].

Additionally, we incorporate feature-matching techniques to ensure the student model captures intermediate representations from the teacher [20].

3.3.1. Response-Based Knowledge Distillation (RKD)

Response-based knowledge distillation (RKD) is a variant of knowledge distillation that refers to the neural response of the last output layer of the teacher model [48]. The operating principle of the RKD is illustrated in Figure 3. According to Figure 3, response-based knowledge focuses on the final output layer of the teacher model. This is accomplished by the assumption that the student model will learn to mimic the predictions of the teacher model.

**Figure 3.** Response-based knowledge distillation [48].

The illustration in Figure 3 shows that this can be achieved by using a loss function, called the distillation loss, which captures the difference between the respective logits of the student model and the teacher model. As this loss would be minimized during the learning process, the student model would become increasingly capable of making the same predictions as the teacher model. By considering the decision-making process of the teacher model, response-based methods can potentially improve the generalization ability and robustness of the student model.

3.3.2. Intermediate Knowledge Distillation

Intermediate-based knowledge distillation (IKD), or feature-based knowledge distillation, is a variant of knowledge distillation in DNN that highlights knowledge learned from hidden layers. The operating principle of IKD is illustrated in Figure 4.

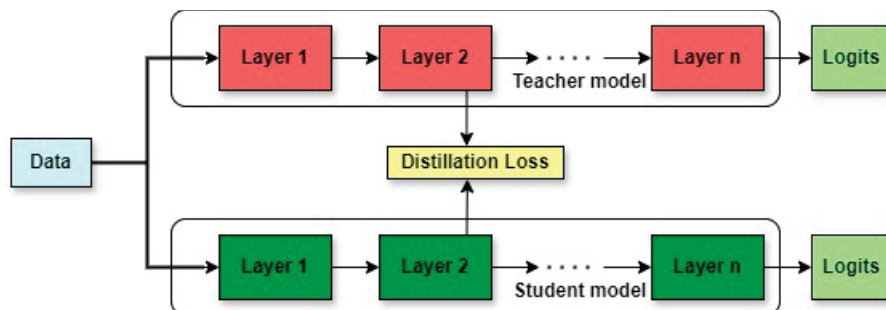


Figure 4. Intermediate knowledge distillation [48].

According to Figure 4, IKD extends traditional knowledge distillation by transferring knowledge not just from the final output layer of the teacher model but also from intermediate layers. Indeed, a trained teacher model also captures knowledge of the data in its intermediate layers, which is particularly relevant for deep neural networks. Thus, the intermediate layers learn to discriminate specific features, and this knowledge can be used to train a student model. As depicted in Figure 4, the aim is to train the student model to learn the same feature activations as the teacher model. The distillation loss function achieves this goal by minimizing the difference between the feature activations of the teacher model and the student model. IKD requires careful design to balance the complexity of transferring knowledge from multiple layers while ensuring computational efficiency and avoiding issues such as vanishing gradients.

4. Experimental Setup and Results Analysis

To investigate the impact of datasets, we conduct experiments with varying configurations, including knowledge distillation with and without dataset-specific adaptations. The success of these manipulations depends on the optimal configuration of experimental parameters and a logical, transparent experimental protocol, which we present in Section 4.1 below.

4.1. Experimental Setup

As shown in Figure 1 and further motivated by the literature review in Section 2, we use teacher-student architecture to distill the knowledge in DNN. So ResNet50 was used as the teacher model and ResNet18 as the student model.

The teacher model is first trained on the original dataset, producing accurate predictions. We also trained the students from scratch to later compare the results after training the students via distillation. Figure 5 shows the validation accuracy over epochs during the training of the teacher and the student from scratch.

During the knowledge distillation process, the student model is trained on the same dataset using a combination of ground truth labels and soft targets generated by the teacher. This dual learning approach helps the student model generalize better and capture intricate patterns. The loss function used in knowledge distillation incorporates both the traditional cross-entropy loss, comparing the student's predictions with the ground truth labels, and a distillation loss, quantifying the similarity between the student's predictions and the soft targets provided by the teacher. The distillation loss encourages the student to mimic the teacher's decision-making process.

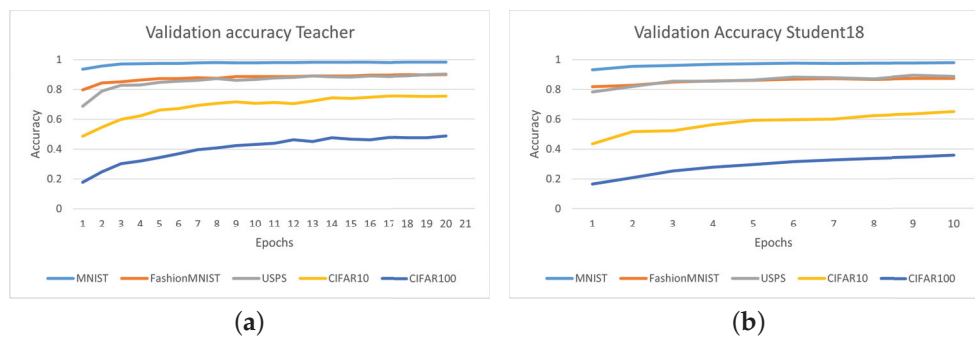


Figure 5. Variation in the validation accuracy by epochs for (a) the teacher model (ResNet50) and (b) the student model (ResNet18).

As ref. [15] confirms that good data augmentation can be used to obtain considerable knowledge distillation. For data augmentation, we use RandomRotation with the value of 15 to randomly rotate the image by up to 15 degrees, RandomHorizontalFlip to randomly flip the image horizontally, and RandomVerticalFlip to randomly flip the image vertically. We transform the images to a PyTorch (version 2.1.2) tensor, and finally, we normalize the data. The cross-entropy loss was used to train all models with the ground truth label, and the distillation loss used was Kullback–Leibler divergence. The hyperparameter controlling the balance between the two losses was $\alpha = 0.7$. The temperature was $t = 4$ [8]. We trained the teacher model within 20 epochs and the students within 10 epochs. We use SGD as an optimizer, and the value of the learning rate was $lr = 0.001$. The Kaggle environment (GPU P100) was used as the hardware and PyTorch as the software to conduct experiments. Each dataset was split into three different subdatasets for training, validation, and testing. The following Table 4 shows the different sizes of each sub-dataset.

Table 4. Distribution of different data sizes for training (83.33%), validation (11.66%), and testing (5%). These data sizes were chosen based on experimental results from the literature review.

Dataset	Training	Validation	Test
CIFAR-10	50,000	7000	3000
CIFAR-100	50,000	7000	3000
USPS	7291	1404	603
MNIST	60,000	7000	3000
Fashion MNIST	60,000	7000	3000

Evaluation metrics encompass traditional classification metrics such as accuracy as shown in Table 1. We conducted multiple runs for each experiment to account for variability and report averaged results for robust conclusions.

Our methodology combines a diverse set of datasets, state-of-the-art model architectures, and a nuanced knowledge distillation process. This comprehensive approach aims to elucidate the impact of datasets on the effectiveness of knowledge distillation in image classification, providing valuable insights for researchers and practitioners in the field.

4.2. Results Analysis

After simulations, the analysis of the results obtained consists in turn of analyzing and comparing the performances of the teacher (ResNet50) and student (Resnet18 from scratch) models on all the data sets (Section 4.2.1). Then, the analysis and comparison of knowledge distilled between the teacher (ResNet50) and the pupil (ResNet18) models in RKD and IKD in Sections 4.2.2 and 4.2.3, respectively.

4.2.1. Analysis of the Results of the Teacher and Student Models from Scratch

Let us remember once again that the first step in knowledge transfer is to train the teacher model since its results will guide the learning of the student model. Figure 6 shows the results of the teacher model after training on the different databases. It also shows the results of the student model from scratch, which will serve as a basis for comparison after knowledge distillation.

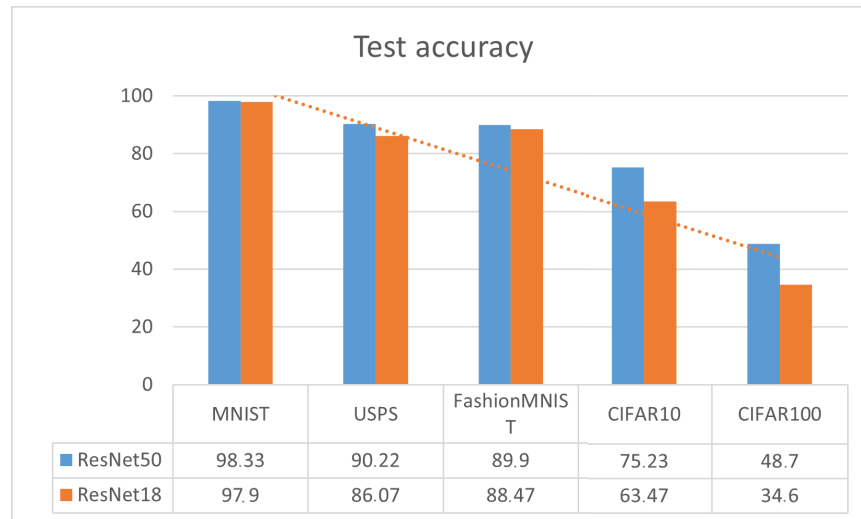


Figure 6. Test accuracy for the teacher and instance student model from scratch.

Looking at Figure 6, we can easily notice that the teacher model performs better than the student model. Indeed, as the student model is shallower than the teacher model, it will also be less accurate. Table 5 completes this figure by presenting the performance differences between the two models on the involved databases. From these representations, we can also see that the performance of both models decreases with database complexity. Further analysis after distillation will enable us to determine whether the same behavior will be observed.

Table 5. Difference between teacher and student accuracy.

Dataset	ResNet50	ResNet18	Difference
MNIST	98.33	97.9	−0.43
FashionMNIST	89.9	88.47	−1.43
USPS	90.22	86.07	−4.15
CIFAR10	75.23	63.47	−11.76
CIFAR100	48.7	34.6	−14.1

Once the teacher model has been trained, the training of the student model can be followed by knowledge distillation. We carried out two different types of distillation experiments, namely RKD [8] for response-based KD and IKD [20] for intermediate-based KD. Sections 4.2.2 and 4.2.3 present the results of these distillations, respectively.

4.2.2. RKD Performance Results Analysis

In the RKD architecture, the student model is trained and guided by the results of the last layer of the teacher model [8]. Figure 7 shows the results of the student model after training by RKD on the different databases.

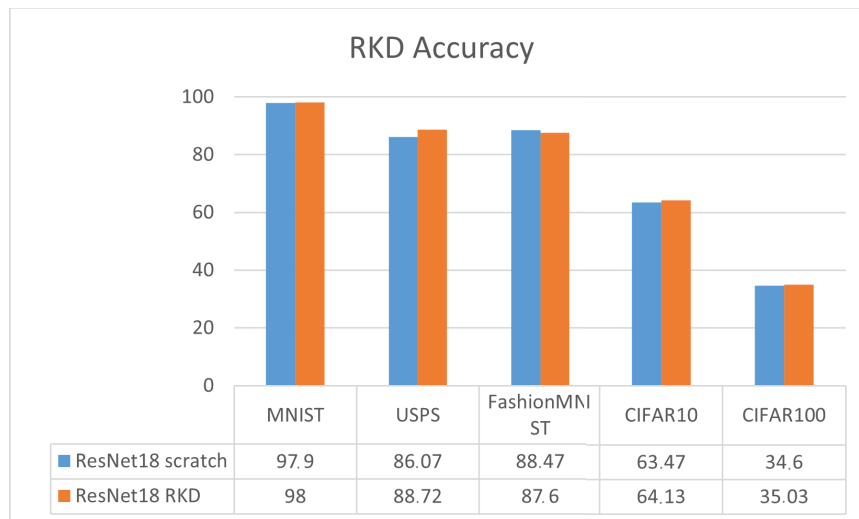


Figure 7. Test accuracy for the teacher and instance student models RKD.

In Figure 7, we can generally see a slight performance gain for the student model. This gain increases as the complexity of the database increases. To complement Figure 7, Table 6 shows the performance gap between the student instance trained from scratch and that trained by response-based knowledge distillation.

Table 6. Difference between the student model from scratch and the student RKD accuracy.

Dataset	ResNet18 Scratch	ResNet18 RKD	Difference
MNIST	97.9	98	+0.1
FashionMNIST	88.47	87.6	−0.87
USPS	86.07	88.72	+2.65
CIFAR10	63.47	64.13	+0.66
CIFAR100	34.6	35.03	+0.43

Part (b) of Figures A1, A3, A5, A7 and A9 shows the precision and loss curves respectively during the epochs of RKD training of the student model on the MNIST, USPS, FashionMNIST, CIFAR10 and CIFAR100 databases.

4.2.3. IKD Performance Results Analysis

In the IKD architecture, the student model is trained and guided by the results of the teacher model's intermediate layer [20]. Figure 8 shows the results of the student model after training by IKD on the different databases.

According to the illustration in Figure 8, we observe a considerable overall performance gain for the student model. This gain is even greater as the complexity of the database increases and is much better than that of RKD. Once again, Table 7 completes Figure 8 by presenting the numerical differences in performance between the student instance trained from scratch and that trained by intermediate-based knowledge distillation (IKD).

Part (c) of Figures A1, A3, A5, A7 and A9 shows the precision and loss curves during the epochs of IKD training of the student model on the MNIST, USPS, FashionMNIST, CIFAR10 and CIFAR100 databases, respectively.

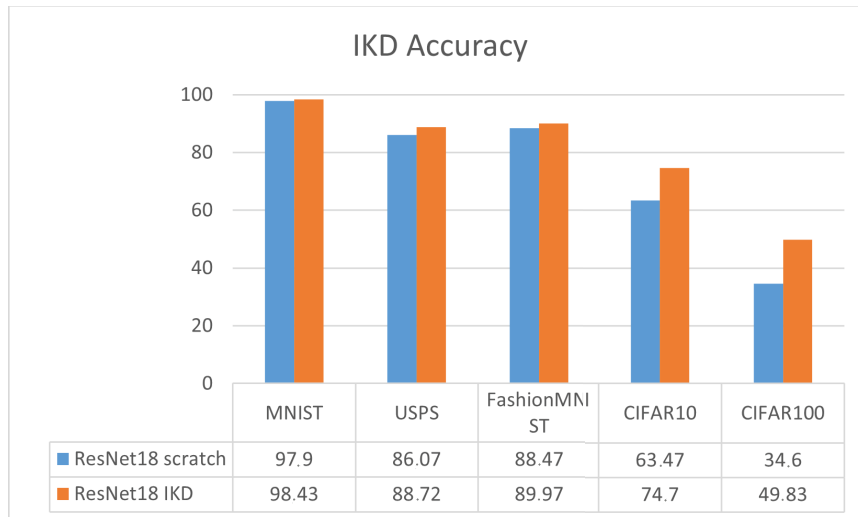


Figure 8. Difference between the student model from scratch and the student IKD accuracy.

Table 7. Table of difference between the student model from scratch and student IKD accuracy.

Dataset	ResNet18 Scratch	ResNet18 IKD	Difference (KD)
MNIST	97.9	98.43	+0.53
FashionMNIST	88.47	89.97	+1.5
USPS	86.07	88.72	+2.65
CIFAR10	63.47	74.7	+10.6
CIFAR100	34.6	49.83	+15.23

4.2.4. Analysis of the Impact of the Database on Knowledge Distillation

After analysing and comparing the results of the teacher model with those of the different instances of the student model, in this section we will analyse the effect of the databases on the distillation itself. To do this, we will first look at Figure 9 which shows the results of the different distillations compared with those of the teacher model; then we will look at Figures 10 and 11 which present the effect of distillation on the different databases and finally we will observe Figure 12 which presents the impact of datasets on knowledge distillation.

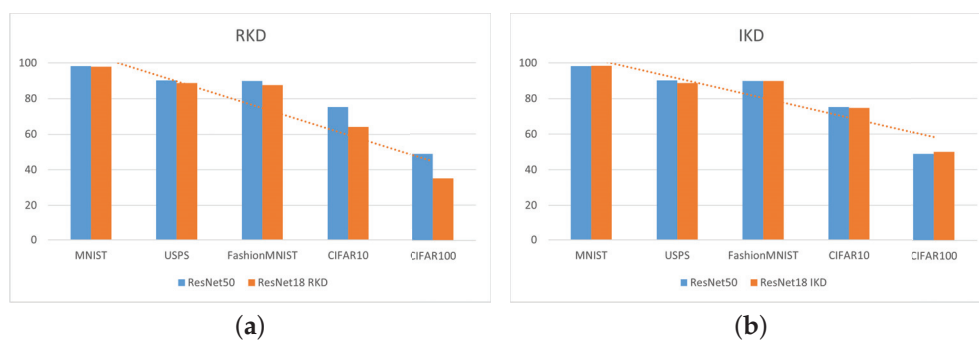


Figure 9. Difference between the teacher and instance student model distilled from RKD (a) and instance student model distilled from IKD (b).

We can draw two major observations from Figure 9 by comparing it with Figure 6. The first observation concerns the RKD: although the student model gains in performance from the RKD, this gain is nevertheless slight, and the observation that the performance of the two models decreases as the complexity of the database increases is confirmed. On the other hand, when we look at the IKD, the gain for the student model is much more

significant. Here, we see that, unlike the others, the student model gains much more in performance as the complexity of the database increases.

Figure 10 shows the gains in student performance after distillation. We first note that IKD [20] performs significantly better than RKD [8]. We note that the more complex the database, the greater the gain in terms of performance. Part (c) IKD of Figures A2, A4, A6, A8 and A10 confirms this last observation. Indeed, we observe a significant increase in the f1-score compared to part (a), from scratch, and part (b), RKD. This increase is proportional to the complexity of the database. We can conclude from this that the more complex the database, the greater the effect of distillation.

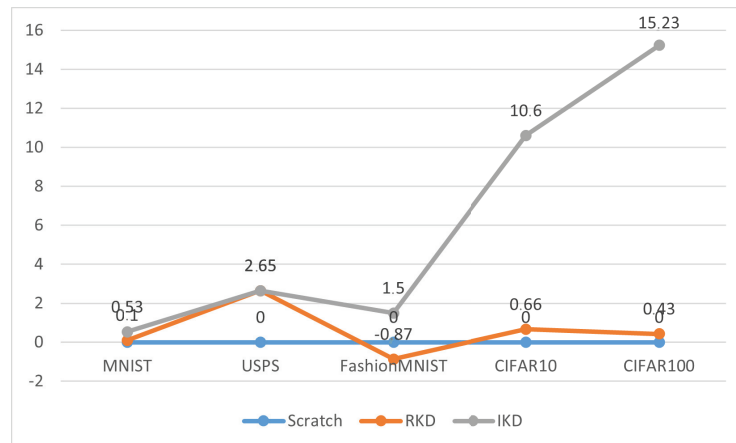


Figure 10. Student performance gain after distillation.

Figure 11 shows us the differences in performance between the different instances of the student model (from scratch, RKD, and IKD) compared to that of the teacher model. Knowledge distillation is indeed effective, and we even note that in the case of IKD, the student performs better than the teacher. On the other hand, we observe that in the least complex databases (MNIST, USPS and FashionMNIST), the performances between the teacher and the different instances of the student are approximately the same. We observe a notable difference in the IKD framework on the CIFAR10 and CIFAR100 databases. This leads us to draw two conclusions:

1. Knowledge distillation has a considerable effect on problems with complex databases. The more complex the database, the deeper and more powerful the model used for training. With a powerful teacher model capable of characterizing knowledge, the transfer to the student model will be assured.
2. By observing the performance provided by RKD and that provided by IKD on different databases, we conclude that the choice of the IKD method will be preferable to that of RKD when dealing with complex databases.

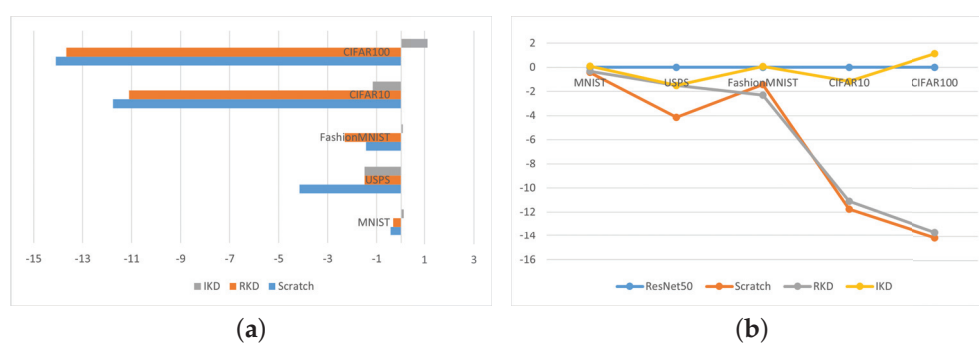


Figure 11. All instances of student performance compared to teacher performance. bar visualisation (a) and curve visualisation (b).

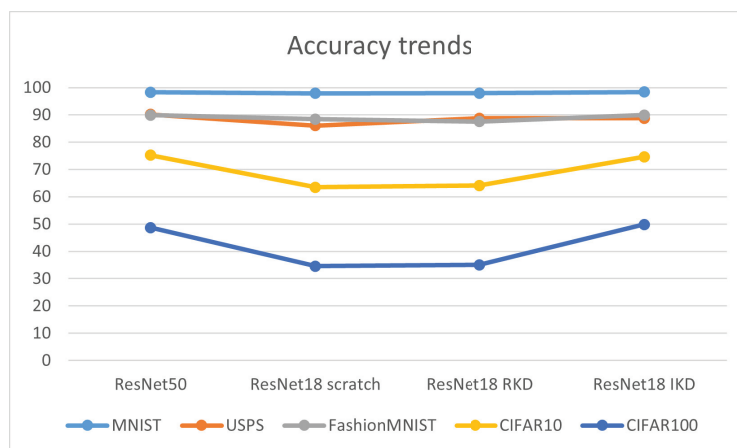


Figure 12. Impact of dataset.

According to Figure 12, we see a slight variation in the curve for the LOW and LOW TO MODERATE databases, namely MNIST and FashionMNIST. The MODERATE (USPS) database curve shows a slightly more marked variation. Finally, the most complex CIFAR10 and CIFAR100 databases (MODERATE TO HIGH and HIGH) show a significant variation.

5. Discussion

The analysis of the results sheds valuable light on the effect of databases on knowledge distillation. By highlighting the importance of choosing the appropriate distillation method according to the complexity of the data and the learning objectives, these results could have important implications for the development of more robust and generalizable learning models. We highlight these insights in Sections 5.1 and 5.2.

5.1. Impact of Database Complexity on Distillation

By examining the performance curves for different databases, we observed significant variations according to the complexity of the data. This observation highlights the importance of considering the diversity of the data and its specific characteristics when designing learning models. The results show a significant difference between RKD and IKD. While RKD shows modest performance gains, IKD shows much more significant improvements, especially with complex databases. This raises questions about the mechanisms underlying these two approaches and their effectiveness in different contexts. More specifically, IKD outperforms RKD, mainly because of the nature of the information that each method transfers from the teacher's model to the student's model. IKD focuses on aligning the student's internal representations or feature maps with those of the teacher at different levels [20,49]. This method ensures that the student model not only learns the final results but also mimics the teacher's hierarchical feature extraction process, capturing richer and more nuanced information throughout its architecture [20]. Indeed, the theoretical underpinnings support this advantage. Intermediate representations contain fine-grained information and hierarchical abstractions that are crucial for complex tasks. By transferring these representations, the student model is better equipped to understand and generalize from the data. This approach exploits the concept of learning intermediate features, which are often more informative than final logs alone, particularly in deep networks where each layer captures progressively higher-level abstractions. In contrast, RKD relies solely on the teacher's final logits [8,9]. Although this method helps the student to know the ultimate limits of the decision, it does not provide the intermediate knowledge essential for a comprehensive understanding of the input [48]. This can lead to less effective transfer, as the student does not benefit from the multi-level learning process followed by the teacher. Interestingly, IKD seems to be more resilient to increasing data complexity; the results show that in some cases, the distilled student (especially with IKD) can even outperform the

teacher in terms of performance. This suggests that the transmission of knowledge through abstract features may be more robust in varied or complex data environments.

That said, knowledge distillation may lead to better generalization or adaptation to specific test data.

5.2. Optimisation of Distillation Strategies

The results indicate the need to develop more sophisticated distillation strategies that take into account the specific nature of the data and the characteristics of the models. In fact, the more complex the database, the greater the effect of distillation on improving the performance of the student model. This observation highlights the importance of taking into account the specific nature of the data when choosing the distillation method and designing the model. According to the results obtained, the IKD method is preferable to RKD due to its greater performance gains.

5.3. Limitation of the Study

Although the results obtained in our work are very interesting, we are aware that our study may have certain limitations. The limited choice of model architecture (ResNet50, ResNet18) used, the fact that the scope was limited to image classification tasks, the nature of the data used, and the choice of distillation methods (RKD, IKD) were deliberate choices to maintain a controlled and detailed analysis in a well-defined context. The performance measures and evaluation methods used in this study could also be a limitation. The scope of the literature search was limited due to access restrictions on some articles, leading to our potentially overlooking important findings that could influence our results. The limitation in isolating the variable impact on knowledge distillation performance; indeed, we compared KD performance on very different datasets rather than systematically varying individual parameters while holding other factors constant. The interpretation of the results is also open to discussion.

6. Conclusions

We conducted a thorough examination of the impact of databases on knowledge distillation in the context of image classification. We have used a diverse array of databases with different levels of complexity. We were able to derive several important and meaningful conclusions by meticulously analyzing the performance of both teacher and student models across various distillation methods.

Firstly, our results clearly demonstrated that knowledge distillation can be pivotal in enhancing the performance of student models, particularly in scenarios where the data are intricate and heterogeneous. Specifically, the IKD method exhibited more substantial performance improvements compared to the RKD method, underscoring the significance of transferring knowledge through abstract and generalizable representations. Furthermore, we observed that the complexity of the database plays a critical role in determining the effectiveness of knowledge distillation. Our findings indicated that as the complexity of the database increases, so do the performance gains of the student model, emphasizing the necessity of considering the unique characteristics of the data during the distillation process.

Additionally, our comprehensive analyses allowed us to compare the performance of the teacher and student models in detail, revealing instances where the distilled student models actually outperformed their teacher counterparts. This observation highlights the remarkable potential of knowledge distillation to foster improved generalization and adaptation to specific test data. Moreover, our results provided guidance on selecting the most appropriate distillation method based on the complexity of the database. Specifically, they suggest that the IKD method is particularly advantageous in scenarios involving complex and varied data.

Overall, our study offers valuable insights into the influence of databases on knowledge distillation, contributing important perspectives for the development of more robust, generalizable, and efficient machine learning models applicable to a wide range of domains. By delving into the nuances of how different distillation methods perform across diverse datasets, we provide a deeper understanding that can inform future research and practical applications in the field of machine learning and image analysis in particular.

Author Contributions: Conceptualization, A.G.B. and C.S.T.K.; methodology, A.G.B. and C.S.T.K.; software, A.G.B. and C.S.T.K.; validation, A.G.B., C.S.T.K., M.E.H. (Mohamed El Haziti) and M.E.H. (Mohammed El Hassouni); data curation, A.G.B. and C.S.T.K.; writing—original draft preparation, A.G.B. and C.S.T.K.; writing—review and editing, A.G.B., C.S.T.K., M.E.H. (Mohamed El Haziti) and M.E.H. (Mohammed El Hassouni); project administration, M.E.H. (Mohamed El Haziti) and M.E.H. (Mohammed El Hassouni); funding acquisition, A.G.B. and C.S.T.K.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset available on request from the authors.

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

Appendix A

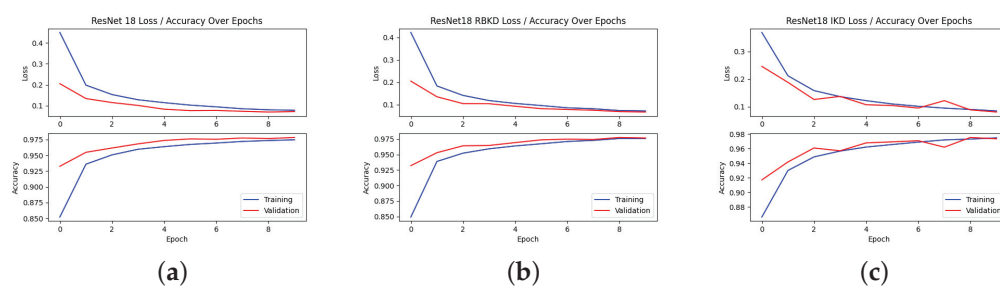


Figure A1. Accuracy and loss over epochs during the training phase of the student model in the MNIST dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

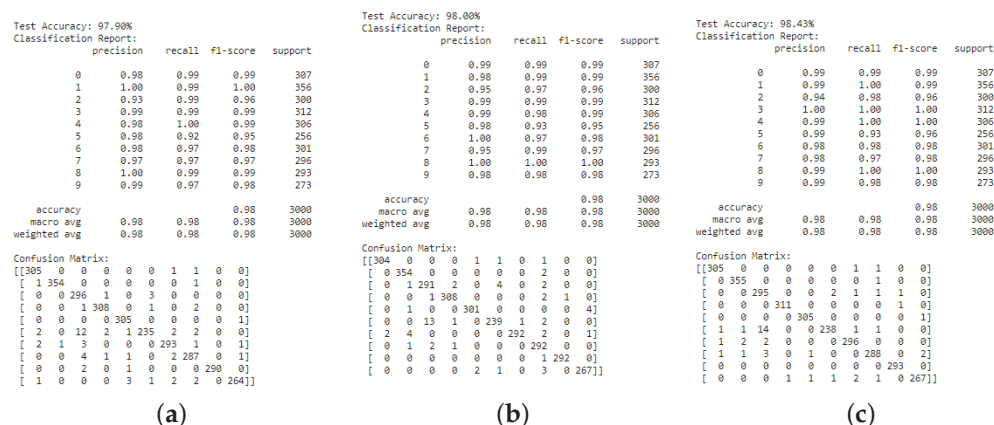


Figure A2. Student metrics after the training phase of the student model in MNIST dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

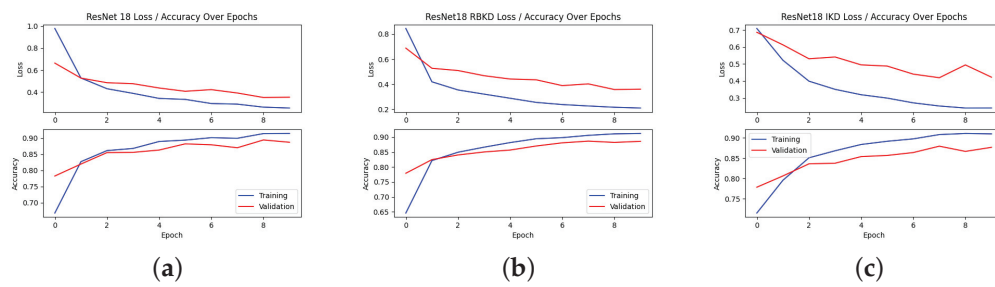


Figure A3. Accuracy and loss over epochs during the training phase of student model in USPS dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

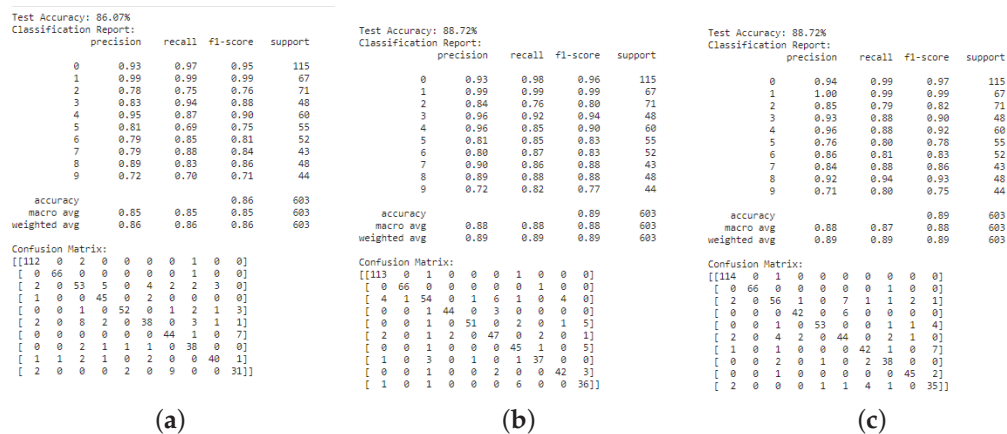


Figure A4. Student metrics after the training phase of the student model in the USPS dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

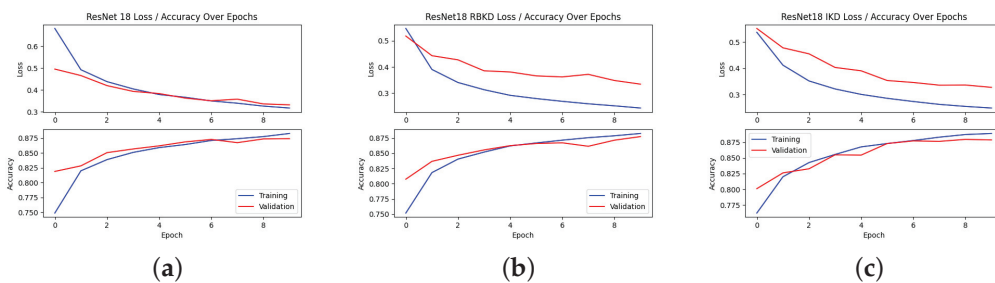


Figure A5. Accuracy and loss over epochs during the training phase of the student model in FashionMNIST dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

Test Accuracy: 88.47% Classification Report:					Test Accuracy: 87.60% Classification Report:					Test Accuracy: 89.97% Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.84	0.82	0.83	311	0	0.84	0.83	0.83	311	0	0.86	0.84	0.85	311
1	0.98	0.97	0.98	300	1	0.99	0.96	0.97	300	1	0.99	0.98	0.98	300
2	0.84	0.77	0.81	311	2	0.83	0.78	0.80	311	2	0.81	0.88	0.84	311
3	0.89	0.91	0.90	320	3	0.87	0.88	0.88	320	3	0.86	0.94	0.90	320
4	0.78	0.82	0.80	322	4	0.75	0.87	0.81	322	4	0.83	0.81	0.82	322
5	0.97	0.97	0.97	301	5	0.97	0.96	0.97	301	5	0.98	0.98	0.98	301
6	0.71	0.72	0.71	311	6	0.72	0.64	0.68	311	6	0.80	0.69	0.74	311
7	0.85	0.84	0.85	278	7	0.94	0.92	0.93	278	7	0.95	0.96	0.95	278
8	0.97	0.97	0.97	263	8	0.96	0.98	0.97	263	8	0.98	0.98	0.98	263
9	0.95	0.98	0.96	283	9	0.93	0.97	0.95	283	9	0.97	0.96	0.97	283
accuracy			0.88	3000	accuracy			0.88	3000	accuracy			0.90	3000
macro avg	0.89	0.89	0.89	3000	macro avg	0.88	0.88	0.88	3000	macro avg	0.90	0.90	0.90	3000
weighted avg	0.89	0.88	0.88	3000	weighted avg	0.88	0.88	0.88	3000	weighted avg	0.90	0.90	0.90	3000

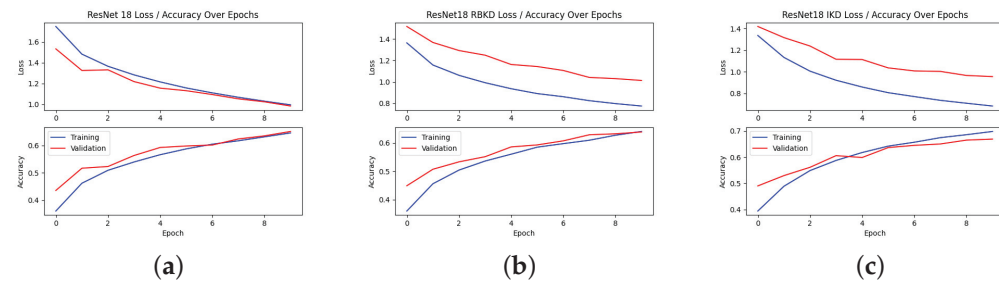
Confusion Matrix:					Confusion Matrix:					Confusion Matrix:				
[[255 1 4 8 0 3 39 0 1 0]]					[[257 0 4 9 2 2 33 1 3 0]]					[[261 1 8 18 1 1 19 0 2 0]]				
[[1 292 0 6 0 0 0 0 1 0]]					[[0 289 0 8 1 0 1 0 1 0]]					[[0 293 0 4 1 0 1 0 1 0]]				
[[5 0 241 5 40 0 20 0 0 0]]					[[4 0 243 4 38 0 22 0 0 0]]					[[4 0 275 5 13 0 14 0 0 0]]				
[[7 1 2 291 9 0 0 9 0 1 0]]					[[10 2 5 282 10 0 11 0 0 0]]					[[4 0 3 301 8 0 4 0 0 0]]				
[[0 2 18 12 265 0 23 0 2 0]]					[[0 1 18 12 279 0 12 0 0 0]]					[[1 3 30 10 262 0 16 0 0 0]]				
[[0 0 0 0 0 292 0 7 0 2]]					[[0 0 0 0 1 0 280 0 7 0 3]]					[[0 0 0 0 0 0 294 0 5 1 1]]				
[[35 0 20 5 25 0 224 0 2 0]]					[[36 1 21 8 40 0 199 0 6 0]]					[[33 0 23 10 29 0 215 0 1 0]]				
[[0 0 0 0 0 5 0 262 0 11]]					[[0 0 0 0 0 5 0 256 0 17]]					[[0 0 0 0 0 5 0 267 0 6]]				
[[0 1 1 0 2 1 0 256 1]]					[[0 0 2 0 1 1 0 0 259 0]]					[[0 0 1 0 1 1 0 258 0]]				
[[0 0 0 0 0 0 0 0 7 0 276]]					[[0 0 0 0 0 1 0 0 0 274]]					[[0 0 0 0 0 0 0 10 0 273]]				

(a)

(b)

(c)

Figure A6. Student metrics after the training phase of the student model in the FashionMNIST dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.



(a)

(b)

(c)

Figure A7. Accuracy and loss over epochs during the training phase of the student model in the CIFAR10 dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

Test Accuracy: 63.47% Classification Report:					Test Accuracy: 64.13% Classification Report:					Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.70	0.67	0.68	294	0	0.63	0.74	0.68	294	0	0.76	0.81	0.79	294
1	0.72	0.77	0.75	293	1	0.72	0.76	0.74	293	1	0.88	0.83	0.86	293
2	0.58	0.55	0.57	303	2	0.52	0.60	0.56	303	2	0.69	0.68	0.68	303
3	0.45	0.37	0.40	300	3	0.45	0.37	0.41	300	3	0.57	0.53	0.55	300
4	0.56	0.59	0.58	303	4	0.56	0.61	0.58	303	4	0.69	0.73	0.71	303
5	0.52	0.49	0.51	301	5	0.59	0.51	0.55	301	5	0.69	0.60	0.64	301
6	0.73	0.76	0.75	308	6	0.75	0.71	0.73	308	6	0.80	0.84	0.82	308
7	0.70	0.62	0.66	293	7	0.74	0.64	0.69	293	7	0.78	0.78	0.78	293
8	0.68	0.77	0.72	291	8	0.76	0.71	0.74	291	8	0.81	0.83	0.82	291
9	0.65	0.75	0.69	314	9	0.68	0.76	0.72	314	9	0.78	0.84	0.81	314
accuracy			0.63	3000	accuracy			0.64	3000	accuracy			0.75	3000
macro avg	0.63	0.63	0.63	3000	macro avg	0.64	0.64	0.64	3000	macro avg	0.75	0.75	0.75	3000
weighted avg	0.63	0.63	0.63	3000	weighted avg	0.64	0.64	0.64	3000	weighted avg	0.75	0.75	0.75	3000

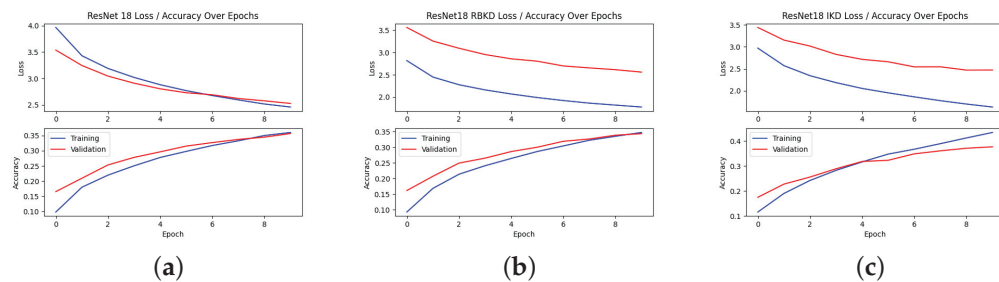
Confusion Matrix:					Confusion Matrix:					Confusion Matrix:				
[[196 13 14 5 5 0 1 4 35 21]]					[[217 8 15 2 5 0 1 2 23 21]]					[[239 5 12 2 8 0 2 3 11 12]]				
[[3 227 2 2 0 1 2 3 9 44]]					[[7 224 7 3 1 1 1 7 41]]					[[1 244 0 2 0 2 3 0 11 30]]				
[[23 2 168 18 32 18 13 11 5]]					[[21 5 181 19 36 12 13 7 6 3]]					[[17 1 207 17 28 7 11 8 1]]				
[[8 5 29 110 29 61 27 9 6 16]]					[[8 8 39 111 22 55 24 13 7 13]]					[[9 2 19 158 21 41 21 10 6 11]]				
[[8 4 25 11 180 22 18 23 9 3]]					[[14 7 30 14 184 8 16 22 3 5]]					[[5 0 24 9 221 9 9 19 6 1]]				
[[5 3 20 57 16 148 14 18 12 8]]					[[7 5 30 50 19 153 8 16 6 7]]					[[6 3 16 57 12 180 7 17 1 2]]				
[[1 2 16 17 19 8 235 2 3 5]]					[[1 4 21 22 21 9 218 1 6 5]]					[[3 0 11 10 12 6 258 1 4 3]]				
[[4 6 11 17 27 19 5 183 5 16]]					[[4 4 18 18 32 13 4 188 1 11]]					[[4 0 7 12 14 13 4 230 2 7]]				
[[24 17 3 2 6 4 1 1 223 10]]					[[47 14 5 2 3 6 0 1 208 5]]					[[19 3 4 6 5 1 4 1 241 7]]				
[[9 35 3 5 5 1 4 4 14 234]]					[[16 32 2 3 4 3 4 4 6 240]]					[[11 19 2 6 0 1 2 4 6 263]]				

(a)

(b)

(c)

Figure A8. Student metrics after the training phase of the student model in the CIFAR10 dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.



(a)

(b)

(c)

Figure A9. Accuracy and loss over epochs during the training phase of the student model in the CIFAR100 dataset. (a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

accuracy	0.35	0.35	0.35	3000	accuracy	0.36	0.35	0.35	3000	accuracy	0.51	0.50	0.50	3000
macro avg	0.35	0.33	0.33	3000	macro avg	0.36	0.35	0.34	3000	macro avg	0.52	0.50	0.49	3000
weighted avg	0.36	0.35	0.34	3000	weighted avg	0.36	0.35	0.34	3000	weighted avg	0.52	0.50	0.49	3000
Confusion Matrix:					Confusion Matrix:					Confusion Matrix:				
[[18 1 0 ... 0 0 0]					[[16 2 0 ... 0 0 0]					[[20 1 0 ... 0 0 0]				
[0 16 0 ... 0 0 0]					[0 17 0 ... 1 0 0]					[0 25 0 ... 0 0 0]				
[0 0 7 ... 0 3 0]					[0 0 8 ... 0 1 0]					[0 0 10 ... 0 2 0]				
...					...					...				
[0 0 0 ... 6 1 0]					[0 0 0 ... 7 0 0]					[0 0 0 ... 14 0 0]				
[0 0 1 ... 0 10 0]					[1 1 0 ... 0 0 0]					[0 0 1 ... 0 9 0]				
[0 0 0 ... 0 0 14]					[0 1 0 ... 0 0 11]					[0 0 0 ... 0 1 19]				

(a)

(b)

(c)

Figure A10. Student metrics after the training phase of the student model in the CIFAR100 dataset.

(a) Training student from scratch, (b) RKD student training, and (c) IKD student training.

References

- Chen, G.; Choi, W.; Yu, X.; Han, T.; Chandraker, M. Learning efficient object detection models with knowledge distillation. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
- Feng, Y.; Wang, H.; Hu, H.R.; Yu, L.; Wang, W.; Wang, S. Triplet distillation for deep face recognition. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Virtual, 25–28 October 2020; pp. 808–812.
- Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; Liu, Q. Tinybert: Distilling bert for natural language understanding. *arXiv* **2019**, arXiv:1909.10351.
- Wang, H.; Li, Y.; Wang, Y.; Hu, H.; Yang, M.H. Collaborative distillation for ultra-resolution universal style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1860–1869.
- Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, Department of Computer Science, University of Toronto, Toronto, ON, Canada, 2009. Available online: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf> (accessed on 1 June 2024).
- LeCun, Y. The MNIST Database of Handwritten Digits. 1998. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 15 July 2024).
- Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747. [CrossRef]
- Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531. [CrossRef]
- Ba, J.; Caruana, R. Do deep nets really need to be deep? *arXiv* **2014**, arXiv:1312.6184.
- Radosavovic, I.; Dollár, P.; Girshick, R.; Gkioxari, G.; He, K. Data distillation: Towards omni-supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4119–4128. [CrossRef]
- Hull, J. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 550–554. [CrossRef]
- Che, Z.; Purushotham, S.; Khemani, R.; Liu, Y. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv* **2015**, arXiv:1512.03542. [CrossRef]
- Tian, Y.; Krishnan, D.; Isola, P. Contrastive representation distillation. *arXiv* **2019**, arXiv:1910.10699. [CrossRef]
- Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31. [CrossRef]
- Wang, H.; Lohit, S.; Jones, M.N.; Fu, Y. What makes a “good” data augmentation in knowledge distillation—a statistical perspective. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 13456–13469.
- Das, D.; Massa, H.; Kulkarni, A.; Rekatsinas, T. An empirical analysis of the impact of data augmentation on knowledge distillation. *arXiv* **2020**, arXiv:2006.03810.
- Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
- Tung, F.; Mori, G. Similarity-preserving knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1365–1374. [CrossRef]
- Alabbasy, F.M.; Abohamama, A.; Alrahmawy, M.F. Compressing medical deep neural network models for edge devices using knowledge distillation. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101616. [CrossRef]
- Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv* **2014**, arXiv:1412.6550. [CrossRef]
- Zhang, X.; Chang, H.; Hao, Y.; Chang, D. MKTN: Adversarial-Based Multifarious Knowledge Transfer Network from Complementary Teachers. *Int. J. Comput. Intell. Syst.* **2024**, *17*, 72. [CrossRef]
- Zhou, T.; Chiam, K.H. Synthetic data generation method for data-free knowledge distillation in regression neural networks. *Expert Syst. Appl.* **2023**, *227*, 120327. [CrossRef]
- Zhang, J.; Tao, Z.; Zhang, S.; Qiao, Z.; Guo, K. Soft Hybrid Knowledge Distillation against deep neural networks. *Neurocomputing* **2024**, *570*, 127142. [CrossRef]
- Wang, C.; Wang, Z.; Chen, D.; Zhou, S.; Feng, Y.; Chen, C. Online adversarial knowledge distillation for graph neural networks. *Expert Syst. Appl.* **2024**, *237*, 121671. [CrossRef]

25. Guermazi, E.; Mdhaftar, A.; Jmaiel, M.; Freisleben, B. MulKD: Multi-layer Knowledge Distillation via collaborative learning. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108170. [CrossRef]
26. Ojha, U.; Li, Y.; Sundara Rajan, A.; Liang, Y.; Lee, Y.J. What knowledge gets distilled in knowledge distillation? *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 11037–11048. [CrossRef]
27. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv* **2016**, arXiv:1612.03928. [CrossRef]
28. Li, H.T.; Lin, S.C.; Chen, C.Y.; Chiang, C.K. Layer-level knowledge distillation for deep neural network learning. *Appl. Sci.* **2019**, *9*, 1966. [CrossRef]
29. Chen, L.; Chen, Y.; Xi, J.; Le, X. Knowledge from the original network: Restore a better pruned network with knowledge distillation. *Complex Intell. Syst.* **2022**, *8*, 709–718. [CrossRef]
30. Yim, J.; Joo, D.; Bae, J.; Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4133–4141. [CrossRef]
31. Srinivas, S.; Fleuret, F. Knowledge transfer with jacobian matching. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4723–4731. Available online: <https://proceedings.mlr.press/v80/srinivas18a.html> (accessed on 25 June 2024).
32. Mirzadeh, S.I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5191–5198. [CrossRef]
33. Bang, D.; Lee, J.; Shim, H. Distilling from professors: Enhancing the knowledge distillation of teachers. *Inf. Sci.* **2021**, *576*, 743–755. [CrossRef]
34. Li, Z.; Li, X.; Yang, L.; Song, R.; Yang, J.; Pan, Z. Dual teachers for self-knowledge distillation. *Pattern Recognit.* **2024**, *151*, 110422. [CrossRef]
35. Shang, R.; Li, W.; Zhu, S.; Jiao, L.; Li, Y. Multi-teacher knowledge distillation based on joint Guidance of Probe and Adaptive Corrector. *Neural Netw.* **2023**, *164*, 345–356. [CrossRef] [PubMed]
36. Cho, Y.; Ham, G.; Lee, J.H.; Kim, D. Ambiguity-aware robust teacher (ART): Enhanced self-knowledge distillation framework with pruned teacher network. *Pattern Recognit.* **2023**, *140*, 109541. [CrossRef]
37. Torralba, A.; Efros, A.A. Unbiased look at dataset bias. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1521–1528.
38. Oliva, A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
39. Basu, M.; Ho, T.K. *Data Complexity in Pattern Recognition*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006. [CrossRef]
40. Shah, B.; Bhavsar, H. Time complexity in deep learning models. *Procedia Comput. Sci.* **2022**, *215*, 202–210. [CrossRef]
41. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
42. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
43. O’shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
44. Sharma, N.; Jain, V.; Mishra, A. An analysis of convolutional neural networks for image classification. *Procedia Comput. Sci.* **2018**, *132*, 377–384. [CrossRef]
45. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef] [PubMed]
46. 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, 21–30 June 2016; pp. 770–778. [CrossRef]
47. Kullback, S. *Information Theory and Statistics*; Courier Corporation: North Chelmsford, MA, USA, 1997.
48. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
49. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [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.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

MDPI Books Editorial Office
E-mail: books@mdpi.com
www.mdpi.com/books



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Topic 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 978-3-7258-9048-4