electronics

Special Issue Reprint

Advances in Social Bots

Edited by
Yangyang Li, Yangzhao Yang and Hu Huang

mdpi.com/journal/electronics

MDPI

Advances in Social Bots

Advances in Social Bots

Guest Editors

Yangyang Li
Yangzhao Yang
Hu Huang

MDPI

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

Guest Editors

Yangyang Li
Key Laboratory of Cyberculture Content Cognition and Detection
University of Science and Technology of China
Hefei
China

Yangzhao Yang
Chinese Institute of Command and Control
Beijing
China

Hu Huang
Shenzhen Graduate School
Peking University
Shenzhen
China

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Electronics* (ISSN 2079-9292), freely accessible at: https://www.mdpi.com/journal/electronics/special_issues/8TF9JML95C.

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

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

ISBN 978-3-7258-6418-8 (Hbk)
ISBN 978-3-7258-6419-5 (PDF)
https://doi.org/10.3390/books978-3-7258-6419-5

Contents

electronics

MDPI

Editorial

Advances in Social Bots

Yangyang Li [1], Yangzhao Yang [1] and Hu Huang [2,*]

[1] Institute of Social Computing, Academy of Cyber, Beijing 100041, China
[2] School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230026, China
* Correspondence: huanghu@ustc.edu.cn

1. Introduction

The digital landscape of the 21st century has been irrevocably shaped by the rise of automated actors. "Social bots"—algorithms designed to generate content and mimic human interaction—have evolved from simple, script-based novelties into sophisticated entities capable of influencing global discourse [1]. While early research primarily viewed these bots as deterministic tools for automated customer service or news aggregation, their capabilities have expanded alongside their potential for misuse. Today, malicious actors are implicated in amplifying low-credibility content, manipulating financial markets, and exacerbating political polarization by infiltrating echo chambers [2,3].

Recent years have witnessed a critical inflection point: the transition from "scripted automation" to "cognitive autonomy". With the advent of Large Language Models (LLMs) and multimodal foundation models, the distinction between human and machine behavior has blurred significantly. Reports indicate that automated traffic surpassed human traffic for the first time in 2024, a shift driven largely by AI-powered agents [4]. Unlike their predecessors, these modern actors possess "emergent abilities"—such as reasoning, planning, and emotional mimicry—allowing them to navigate complex social dynamics with unprecedented fluidity [5,6].

This evolution has fundamentally altered the nature of online interaction. On an individual level, modern agents can now engage in long-term strategic planning and exhibit persuasive capabilities that rival human interlocutors. Studies suggest that LLM-driven agents can tailor rhetoric to specific user demographics, rendering them potent tools for computational propaganda [7]. On a collective level, these agents are increasingly deployed in "social sandboxes" to simulate human community dynamics. While offering a new lens for computational sociology, this also raises concerns about the scalability of synthetic misinformation [8].

Consequently, authenticating these interactions is becoming increasingly difficult as the "detection boundary" shifts. Traditional models often fail to capture the long-range dependencies inherent in multi-party, multi-turn discussions, a challenge exacerbated by the scarcity of realistic conversational datasets [9]. Furthermore, the ability of agents to utilize external tools (e.g., search engines, APIs) allows them to ground responses in real-time data. This capability enables them to bypass detection methods that rely on identifying factual hallucinations or static knowledge cutoffs [10].

We are thus witnessing a paradigm shift from disembodied software scripts to "Social Agents"—autonomous systems capable of constructing internal world models, maintaining long-term memory, and interacting with the physical world through Embodied AI [11]. However, realizing this vision requires a holistic approach. To build agents that can effectively persuade without hallucinating, or navigate physical spaces without failing, we must address fundamental challenges in intent understanding, environmental grounding,

and resilient connectivity. This Special Issue, "Advances in Social Bots," was curated to bridge this gap, providing the architectural blueprints that span from cognitive algorithms to the necessary physical infrastructure.

2. Thematic Overview: From Cognition to Infrastructure

The articles in this Special Issue illustrate that the evolution from "scripted bots" to "autonomous agents" is not a single leap, but a layered evolution. To support the high-level emergent behaviors described in the Introduction—such as strategic persuasion and adaptive social interaction—advancements are required across three fundamental layers: cognitive processing, perceptual adaptation, and physical infrastructure.

2.1. Cognitive Intelligence: Understanding Stance and Sentiment

For a social agent to interact meaningfully, it must understand not just what is said, but the stance and sentiment behind it.

Xie et al. (Contribution 1) address the challenge of zero-shot stance detection. They propose the PAMR (Pragmatic-Aware Multi-Agent Reasoning) framework, which utilizes LLMs in a multi-agent architecture. By explicitly modeling pragmatic cues like sarcasm, their work demonstrates how agents can "think" collaboratively to decipher implicit stances without task-specific training. Similarly focusing on linguistic nuance, Xu et al. (Contribution 2) explore the granularity of emotion. Their research on aspect-level sentiment analysis allows machines to disentangle complex sentence structures, a critical capability for agents engaging in nuanced human–machine dialogue.

Furthermore, Zeng et al. (Contribution 3) tackle the temporal dimension. They propose the DASR framework for joint event detection, utilizing incremental learning to mitigate "catastrophic forgetting." This ensures that agents can continuously adapt to emerging hot topics, mitigating the risk of knowledge obsolescence in long-term social simulations.

2.2. Perceptual Robustness: Vision and Adaptation

Modern agents operate in a multimodal world. As the boundary between real and synthetic content blurs, perceptual robustness becomes critical—both for agents to ground themselves in reality and for systems to distinguish authentic interactions from fabricated ones.

Chen et al. (Contribution 4) explore the intersection of vision and geography. Their FLsM model utilizes large-scale visual models for the fuzzy localization of image scenes. This capability is vital for verifying the authenticity of user-generated content and distinguishing between real-world activity and fabricated bot personas. Addressing the stability of AI models in changing environments, Song et al. (Contribution 5) introduce a "Dual Constraints" method for Continual Test-Time Adaptation (CTTA). This research ensures that detection algorithms remain robust even as data distributions shift over time, providing a defense against bots that constantly alter their behavioral patterns to evade detection.

2.3. Embodied Intelligence and Network Infrastructure

Finally, as agents transition from digital chatbots to embodied robots interacting with the physical world, they require sensitive perception and resilient communication networks to maintain autonomy.

Sharma et al. (Contribution 6) provide a glimpse into embodied interaction. Their work on hardness classification using cost-effective tactile sensors mimics human mechanoreceptors, paving the way for service robots that can socially and physically interact with humans.

Supporting these distributed agents requires robust connectivity. Qu et al. contribute two pivotal studies on the network layer. In their first paper (Contribution 7), they optimize the geometry of Stratospheric Pseudolite Networks (SPNs). In their second paper (Contribution 8), they propose an Intelligent Pseudolite Constellation (IPCB) based on high-altitude balloons. These studies lay the foundation for a resilient, wide-coverage communication infrastructure. Such networks are essential for coordinating swarms of autonomous agents in remote areas where terrestrial networks fail, ensuring that the "social" connection remains unbroken.

3. Future Outlook

The research presented in this Special Issue highlights an inevitable trajectory: the convergence of Generative AI, robotics, and social computing. We are witnessing a paradigm shift from simple "Social Bots" to sophisticated "Social Agents"—entities that can think, see, and touch. Unlike their text-centric predecessors, these future agents will operate across three interconnected dimensions, as evidenced by the contributions in this volume.

- Cognitive Autonomy: Future agents must move beyond script adherence to exhibit emergent behaviors and reasoning. They will simulate complex human societal dynamics in computational sandboxes, a direction supported by recent surveys on LLM-based multi-agent systems [12].
- Multimodal Perception: Agents will seamlessly transition between digital platforms and physical forms. They will require visual adaptability to verify reality and tactile sensitivity to interact with the physical world.
- Resilient Infrastructure: The sustainability of these agents will depend on robust network architectures capable of supporting distributed, autonomous swarms in even the most remote environments.

As these technologies mature, the challenge for the scientific community shifts from merely detecting these actors to understanding their complex interactions with human society. The boundary between human and machine is blurring, necessitating new governance frameworks to ensure that these powerful agents align with human values.

4. Conclusions

We extend our gratitude to all the authors, reviewers, and the editorial team who made this Special Issue possible. "Advances in Social Bots" stands as a testament to the field's diversity, successfully bridging the gap between abstract software algorithms and tangible hardware engineering.

By integrating cognitive intelligence, perceptual robustness, and physical infrastructure, this collection provides the foundational blueprints for the next generation of autonomous systems. We hope this Special Issue inspires further inquiry into the symbiotic future of human and machine intelligence.

Funding: The work was partly supported by the National Natural Science Foundation of China (U25B2042).

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

List of Contributions

1. Xie, Z.; Niu, F.; Dai, G.; Zhang, B. From Claims to Stance: Zero-Shot Detection with Pragmatic-Aware Multi-Agent Reasoning. *Electronics* **2025**, *14*, 4298. https://doi.org/10.3390/electronics14214298.
2. Xu, E.; Zhu, J.; Zhang, L.; Wang, Y.; Lin, W. Research on Aspect-Level Sentiment Analysis Based on Adversarial Training and Dependency Parsing. *Electronics* **2024**, *13*, 1993. https://doi.org/10.3390/electronics13101993.

3. Zeng, X.; Luo, G.; Qin, K. Joint Event Detection with Dynamic Adaptation and Semantic Relevance. *Electronics* **2025**, *14*, 234. https://doi.org/10.3390/electronics14020234.
4. Chen, W.; Miao, L.; Gui, J.; Wang, Y.; Li, Y. FLsM: Fuzzy Localization of Image Scenes Based on Large Models. *Electronics* **2024**, *13*, 2106. https://doi.org/10.3390/electronics13112106.
5. Song, Y.; Liu, P.; Wu, Y. Enhancing the Sustained Capability of Continual Test-Time Adaptation with Dual Constraints. *Electronics* **2025**, *14*, 3891. https://doi.org/10.3390/electronics14193891.
6. Sharma, Y.; Ferreira, P.; Justham, L. Hardness Classification Using Cost-Effective Off-the-Shelf Tactile Sensors Inspired by Mechanoreceptors. *Electronics* **2024**, *13*, 2450. https://doi.org/10.3390/electronics13132450.
7. Qu, Y.; Wang, S.; Feng, H.; Liu, Q. Geometry Optimization of Stratospheric Pseudolite Network for Navigation Applications. *Electronics* **2024**, *13*, 2397. https://doi.org/10.3390/electronics13122397.
8. Qu, Y.; Wang, S.; Pan, T.; Feng, H. IPCB: Intelligent Pseudolite Constellation Based on High-Altitude Balloons. *Electronics* **2024**, *13*, 2095. https://doi.org/10.3390/electronics13112095.

References

1. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The rise of social bots. *Commun. ACM* **2016**, *59*, 96–104. [CrossRef]
2. Shao, C.; Ciampaglia, G.L.; Varol, O.; Yang, K.C.; Flammini, A.; Menczer, F. The spread of low-credibility content by social bots. *Nat. Commun.* **2018**, *9*, 4787. [CrossRef] [PubMed]
3. Cresci, S. A decade of social bot detection. *Commun. ACM* **2020**, *63*, 72–83. [CrossRef]
4. Thales Group. 2025 Bad Bot Report. 2025. Available online: https://www.imperva.com/resources/resource-library/reports/2025-bad-bot-report/ (accessed on 24 November 2025).
5. Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative Agents: Interactive Simulacra of Human Behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST'23), New York, NY, USA, 29 October–1 November 2023. [CrossRef]
6. Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. The rise and potential of large language model based agents: A survey. *Sci. China Inf. Sci.* **2025**, *68*, 121101. [CrossRef]
7. Salvi, F.; Ribeiro, M.H.; Gallotti, R.; West, R. On the conversational persuasiveness of large language models: A randomized controlled trial. *arXiv* **2024**, arXiv:2403.14380. [CrossRef]
8. Gao, C.; Lan, X.; Lu, Z.; Mao, J.; Piao, J.; Wang, H.; Jin, D.; Li, Y. S3: Social-network simulation system with large language model-empowered agents. *arXiv* **2023**, arXiv:2307.14984. [CrossRef]
9. Niu, F.; Yang, M.; Li, A.; Zhang, B.; Peng, X.; Zhang, B. A Challenge Dataset and Effective Models for Conversational Stance Detection. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italy, 20–25 May 2024. [CrossRef]
10. Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; Anandkumar, A. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv* **2023**, arXiv:2305.16291. [CrossRef]
11. Fung, P.; Bachrach, Y.; Celikyilmaz, A.; Chaudhuri, K.; Chen, D.; Chung, W.; Dupoux, E.; Gong, H.; Jégou, H.; Lazaric, A.; et al. Embodied ai agents: Modeling the world. *arXiv* **2025**, arXiv:2506.22355. [CrossRef]
12. Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, Jeju, Republic of Korea, 3–9 August 2024; pp. 8048–8057.

electronics

MDPI

Article

From Claims to Stance: Zero-Shot Detection with Pragmatic-Aware Multi-Agent Reasoning

Zhiyu Xie [1,†], Fuqiang Niu [2,†], Genan Dai [1] and Bowen Zhang [1,*]

[1] School of Artificial Intelligence, Shenzhen Technology University, Shenzhen 518118, China
[2] School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230026, China
* Correspondence: zhangbowen@sztu.edu.cn
† These authors contributed equally to this work.

Abstract: Stance detection aims to identify whether a text expresses a favorable, opposing, or neutral attitude toward a given target and has become increasingly important for analyzing public discourse on social media. Existing approaches, ranging from supervised neural models to prompt-based large language models (LLMs), face two persistent challenges: the scarcity of annotated stance data across diverse targets and the difficulty of generalizing to unseen targets under pragmatic and rhetorical variation. To address these issues, we propose PAMR (Pragmatic-Aware Multi-Agent Reasoning), a zero-shot stance detection framework that decomposes stance inference into structured reasoning steps. PAMR orchestrates three LLM-driven agents—a linguistic parser that extracts pragmatic markers and canonicalizes claims, an NLI-based estimator that produces calibrated stance probabilities through consensus voting, and a counterfactual and view-switching auditor that probes robustness under controlled rewrites. A stability-aware fusion integrates these signals, conservatively abstaining when evidence is uncertain or inconsistent. Experiments on SemEval-2016 and COVID-19-Stance show that PAMR achieves macro-F1 scores of 71.9% and 73.0%, surpassing strong zero-shot baselines (FOLAR and LogiMDF) by +2.0% and +3.1%. Ablation results confirm that pragmatic cues and counterfactual reasoning substantially enhance robustness and interpretability, underscoring the value of explicit reasoning and pragmatic awareness for reliable zero-shot stance detection on social media.

Keywords: zero-shot stance detection; multi-agent framework; large language model

1. Introduction

Social media has become a central arena for public discourse, where individuals express their opinions on controversial issues such as politics, social movements, healthcare, and environmental policies. Understanding public stance toward such topics is crucial for applications including opinion mining, misinformation detection, crisis monitoring, and policy-making support. Stance detection—the task of classifying a text as favorable, against, or neutral with respect to a target—has therefore emerged as a fundamental problem in natural language processing (NLP) and social computing [1]. Early studies treated stance detection as a supervised classification task on social media platforms such as Twitter, focusing on political debates and event-specific targets [2,3]. Subsequent research leveraged neural architectures and pre-trained language models to improve contextual understanding [4,5], yet most methods still depend on large annotated datasets and struggle to generalize across unseen targets. These limitations have motivated recent exploration into cross-target and zero-shot stance detection paradigms [6,7], which aim to infer stance for new topics without task-specific training data. In practice, however, stance on social

media is rarely expressed in a straightforward manner. Instead, it is often conveyed implicitly through pragmatic cues such as sarcasm, negation, rhetorical questions, or figurative language [8]. These linguistic devices can obscure or even invert the intended polarity, making it difficult for systems to distinguish stance from surface sentiment [9]. This suggests that effective stance detection should not treat the input merely as a text–target pair, but rather as a reasoning task that accounts for the author's underlying claim and its pragmatic framing.

Traditional stance detection methods initially focused on supervised neural architectures, including recurrent networks, attention mechanisms, and graph-based models, which rely on large quantities of annotated stance data [3,4,10,11]. With the rise of pre-trained language models (PLMs), fine-tuning strategies achieved stronger performance by leveraging contextual embeddings [5,12]. However, these methods still face two fundamental challenges. First, annotated stance data remain scarce and unevenly distributed across targets, making supervised training impractical for new domains. Second, generalization to unseen targets remains difficult, even with powerful PLMs, as stance expressions often vary in subtle ways across topics [13].

To alleviate these issues, recent research has turned to cross-target [6] and zero-shot stance detection (ZSSD) [7]. Cross-target methods attempt to transfer stance knowledge from source domains to unseen targets using contrastive learning, knowledge injection, or graph reasoning. Meanwhile, zero-shot approaches leverage PLMs and large language models (LLMs) directly via prompting or inference reformulation [14]. Despite recent progress, two persistent gaps remain. First, many approaches conflate stance with sentiment, especially in emotionally charged or figurative language. Second, existing models lack explicit mechanisms to handle pragmatic phenomena—such as sarcasm, hedging, and negation—or to assess whether stance predictions are stable under slight linguistic variations. These challenges suggest that monolithic architectures may struggle to disentangle the complex and often interacting cues that shape stance expression.

These observations suggest several concrete desiderata for zero-shot stance detection systems: (1) extract explicit, target-linked canonical claims rather than relying solely on surface expressions; (2) remain robust to pragmatic confounds such as sarcasm and negation; (3) ensure prediction stability through counterfactual and perspective-based probing; (4) adopt calibrated decision strategies that abstain to neutral when evidence is thin or unstable.

Inspired by recent advances in multi-agent reasoning—particularly division of labor, coordination, and self-verification capabilities [15,16], we introduce **PAMR**—Pragmatic-Aware Multi-Agent Reasoning—a zero-shot stance detection framework that decomposes the task into interpretable subtasks and re-assembles their signals through stability-aware decision making. PAMR orchestrates three LLM-driven agents: (1) a Linguistic Parser that distills the input into a canonical claim while extracting pragmatic markers; (2) an NLI-based Estimator that produces a calibrated distribution {favor, against, neutral}; and (3) a Counterfactual and View-Switching module that probes robustness by re-evaluating stance under meaning-preserving rewrites (e.g., removing sarcasm, switching voice). A lightweight stability-aware fusion integrates these signals, conservatively assigning neutral when confidence is low, predictions are unstable, or top classes are tied. PAMR requires no task-specific fine-tuning, yields auditable intermediate outputs (claims, pragmatic markers, robustness flips), and explicitly mitigates common failure modes such as sarcasm-induced polarity errors.

The main contributions of this paper are summarized as follows:

1. We propose PAMR, a pragmatic-aware multi-agent framework for zero-shot stance detection that factors inference into claim normalization, probabilistic NLI, and robustness probing, enabling interpretable and modular reasoning.
2. We introduce a counterfactual and view-switching probe that quantifies stability of stance under meaning-preserving edits and perspective shifts; this signal directly informs a stability-aware fusion rule that curbs over-confident polarity errors and disentangles stance from sentiment.
3. We conduct extensive experiments on benchmark datasets, demonstrating that PAMR matches or surpasses strong zero-shot baselines. Ablations reveal the contributions of pragmatic cues and robustness probing, while additional analyses show that PAMR produces interpretable intermediate artifacts that enable fine-grained audits.

Overall, this study bridges computational modeling and social media discourse analysis by explicitly integrating pragmatic reasoning into stance inference. We aim to provide a framework that not only improves zero-shot prediction accuracy but also deepens interpretability for real-world applications such as misinformation monitoring and public opinion tracking. The remainder of this paper is organized as follows. Section 2 introduces related work. Section 3 describes the proposed method. Section 4 presents the experimental setup. Section 5 reports the experimental results and analysis. Section 6 discusses key findings and limitations. Section 7 concludes the paper.

2. Related Works

2.1. Traditional Stance Detection Methods

Stance detection, closely related to sentiment analysis, argument mining, and fact verification, has been studied extensively in natural language processing. Early research primarily relied on supervised learning with handcrafted features and classical classifiers. With the development of deep learning, neural architectures such as CNNs [17,18] and LSTMs [10] became prevalent, enabling models to capture contextual and sequential dependencies in text. Graph neural networks were later introduced to encode relations among posts, users, and targets, further enriching stance representations. The emergence of PLMs significantly advanced the field. Fine-tuning strategies reformulated stance detection as a text–target classification problem by concatenating target and input sequences [19]. To reduce computational cost, lightweight adaptation methods froze most PLM parameters while tuning small modules [14]. Beyond fine-tuning, prompt-based techniques framed stance detection as masked language modeling. By filling in templates such as "The attitude toward <Target> is [MASK]," PLMs could predict stance more effectively. Recent studies improved this paradigm by designing adaptive prompts tailored to different targets [20]. In addition to general PLMs, social-media-oriented variants such as BERTweet [21] and CT-BERT [22] have been developed to better capture the linguistic characteristics of Twitter and COVID-19 discourse. While these models enhance representation learning on noisy text, they remain limited in addressing pragmatic phenomena such as sarcasm or negation and lack interpretability or stability validation mechanisms. Multi-modal variants have also been proposed, combining textual and visual cues through prompt-based mechanisms [23]. Despite these innovations, most traditional approaches remain reliant on annotated supervision and show limited robustness when confronted with pragmatic language use.

2.2. Zero-Shot Stance Detection

To overcome the dependence on labeled data, zero-shot stance detection has been widely explored. In this setting, models must transfer knowledge from source targets to

entirely new ones without direct supervision. Early work employed contrastive learning to align stance features across domains, as in JointCL [24], while other approaches leveraged external knowledge bases to bridge semantic gaps between known and unseen targets, such as TarBK. Graph-based methods further advanced the field by constructing heterogeneous or multi-view graphs that capture transferable stance signals among tweets, claims, and targets. Recently, researchers have investigated reasoning-based enhancements to improve zero-shot generalization. LogiMDF [25] integrates first-order logic constraints into multi-decision fusion, ensuring consistency across LLM outputs while leveraging hypergraph propagation. These methods highlight a shift from purely data-driven transfer to structured reasoning, which improves both robustness and explainability. Nevertheless, pragmatic challenges such as sarcasm and rhetorical framing remain largely unresolved in ZSSD.

2.3. LLM-Based Stance Detection Methods

The remarkable zero-shot and few-shot capabilities of large language models have reshaped stance detection research. One line of work directly treats LLMs as stance predictors: with carefully designed prompts, LLMs can classify stance without additional training [14,26]. Strategies include direct zero-shot prompting, chain-of-thought reasoning, and prompt designs enriched with background knowledge. While flexible, such direct use often leads to unstable outputs and inconsistent predictions. Another line of research employs LLMs as knowledge providers that augment smaller, trainable models. For example, domain-relevant background information can be elicited from LLMs and injected into stance classifiers [27], combining the interpretive capacity of LLMs with the adaptability of supervised models. More recent efforts emphasize structured reasoning with interpretable intermediate steps. Enhanced CoT methods [28–30] decompose stance prediction into step-wise inferences (e.g., from factual entailment to subjective alignment), but rarely define rule-based constraints or auditable outputs. Some systems, such as FOLAR, integrate symbolic representations like sentiment trajectories or discourse role tags, while LogiMDF incorporates logical rules over hypergraphs for traceable reasoning. However, the internal reasoning paths often remain implicit or non-verifiable.

2.4. Summary and Research Gap

Across these paradigms, most prior studies emphasize semantic transfer or large-scale contextual modeling but seldom incorporate explicit pragmatic reasoning or stability verification. For instance, social-media-oriented models such as BERTweet, CT-BERT, and RoBERTa-Twitter capture linguistic variations in tweets but remain limited in handling pragmatic phenomena like sarcasm or negation. Reasoning-based approaches (e.g., LogiMDF, CIRF) introduce logical or cognitive structures, yet their reasoning traces are often implicit and lack auditable intermediate outputs. Overall, existing frameworks can be contrasted along three dimensions—(1) whether pragmatic cues are explicitly modeled, (2) whether prediction stability is verified, and (3) whether interpretable intermediate reasoning is provided. PAMR addresses these gaps by integrating all three: it explicitly encodes pragmatic markers, employs counterfactual probing for stability, and yields interpretable intermediate artifacts for transparent stance reasoning.

3. Methods

3.1. Task Definition

Given a short text x and a target t, the goal of zero-shot stance detection is to predict

$$y \in \{\texttt{favor}, \texttt{against}, \texttt{neutral}\}$$

without using any annotated stance data for t, i.e., $\mathcal{T}_{\text{train}} \cap \mathcal{T}_{\text{test}} = \emptyset$. Our approach decomposes this prediction into multi-agent reasoning steps with interpretable intermediate outputs and evaluates prediction stability via agent-level sampling.

3.2. PAMR Overview

As shown in Figure 1, PAMR (*Pragmatic-Aware Multi-Agent Reasoning*) decomposes zero-shot stance detection into four interpretable stages, each implemented as an explicit agent that transforms inputs into structured intermediate representations. These agents interact in a modular pipeline to progressively refine stance decisions.

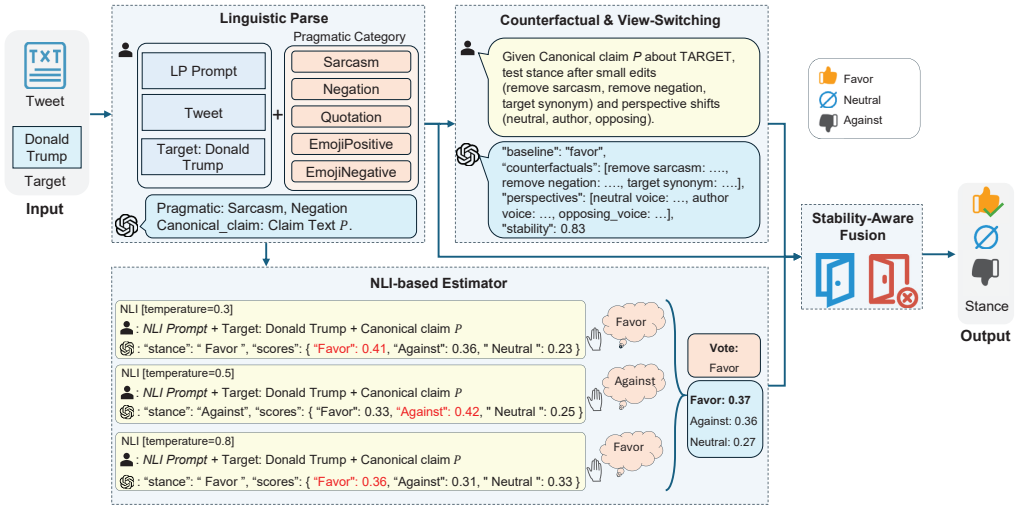


Figure 1. PAMR Framework overall.

(1) **Linguistic Parse Agent**: takes the raw social media post x and target t as input, and outputs a *target-linked canonical claim* along with pragmatic markers such as sarcasm, hedging, or negation. This ensures that subsequent reasoning is grounded in normalized propositions instead of noisy surface forms.

(2) **NLI Estimation Agent**: Reformulates stance classification as a natural language inference task. Given the canonical claim and target, it runs multiple inference passes using diverse prompts to produce both a probability distribution over stance labels and a consensus vote count, mitigating randomness.

(3) **Counterfactual and View-Switching Agent**: Evaluates the stability of the predicted stance by applying minimal, meaning-preserving perturbations (e.g., changing tone, perspective). It outputs a scalar *stability score* reflecting how robust the original decision is under linguistic variation.

(4) **Stability-Aware Fusion Agent**: Receives all signals—canonical claim, pragmatic tags, NLI probabilities and consensus, stability score—and integrates them under a conservative policy. It abstains to "neutral" if confidence is low or predictions are unstable, ensuring robustness.

Each agent has a well-defined role, input–output interface, and interpretable output, and the full pipeline corresponds directly to the annotated modules in Figure 1. In particular, the Linguistic Parse Agent is responsible for the canonicalization and pragmatic enrichment mentioned in the figure.

3.3. Linguistic Parse

The Linguistic Parse Agent aims to normalize noisy social media text into a target-linked canonical claim, while detecting pragmatic markers (e.g., sarcasm, negation, quota-

tion, emoji) that may invert or obscure stance polarity. Given a tweet x and target t, the LLM generates a structured output containing pragmatic tags and a canonical claim c.

Canonical Claim Extraction: The model distills the author's underlying proposition regarding the target into a clear canonical claim, ensuring that the subsequent stance reasoning operates on an explicit statement rather than noisy surface text.

Pragmatic Tagging: The parser identifies pragmatic markers such as sarcasm, negation, quotation, and emoji usage, which are later incorporated in fusion to calibrate stance decisions.

Input Data: Tweet x, Target t
Prompt: Identify pragmatic markers (sarcasm, negation, quotation, emoji) and rewrite the tweet into a canonical claim about the target.
Expected Output: Pragmatic tags and a canonical claim c.

3.4. NLI-Based Estimator

The NLI-based Estimator stage reframes stance detection as a natural language inference (NLI) problem. The canonical claim c and target t are paired as premise–hypothesis inputs to the LLM. To mitigate stochasticity and improve reliability, the model performs n independent inference runs under different decoding parameters.

Multi-run Voting: Each run outputs a stance label $\{favor, against, neutral\}$ and a probability distribution $\mathbf{p}$. A majority vote and averaged probability vector $\bar{\mathbf{p}}$ are then aggregated for stability.

Input Data: Canonical claim c, Target t
Prompt: Given c and t, determine whether the author's stance is favor, against, or neutral. Output both the label and probability scores.
Expected Output: $\{stance, scores\}$ per run, aggregated into majority label $y^\star$, vote ratio r, and averaged scores $\bar{\mathbf{p}}$.

3.5. Counterfactual and View-Switching

The Counterfactual and View-Switching (CVS) stage evaluates robustness of stance predictions by rephrasing the canonical claim under minimal edits and perspective shifts. This stage probes whether stance remains consistent across semantically faithful variations.

Counterfactual Edits: The claim is modified by (i) removing sarcasm, (ii) removing negation, and (iii) replacing the target with a synonym. Perspective Shifts: The claim is paraphrased into three perspectives: neutral voice, author voice, and opposing voice. Stability Score: The proportion of paraphrases consistent with the baseline stance is reported as the stability score S. By definition, the stability score $S \in [0, 1]$ quantifies the proportion of paraphrases whose predicted stance remains consistent with the baseline. Higher values of S therefore indicate greater prediction stability under controlled linguistic rewrites, and thus higher confidence in the stance.

Input Data: Canonical claim c, Target t
Prompt: Given c, re-evaluate stance after small edits (remove sarcasm, remove negation, target synonym) and perspective shifts (neutral, author, opposing).
Expected Output: JSON with baseline stance, stance for each edit/perspective, and stability score S.

3.6. Stability-Aware Fusion

Thresholds such as $\tau_{\text{unstable}} = 0.40$ control model behavior; for example, if $S < \tau_{\text{unstable}}$, the model abstains by predicting "neutral" to avoid over-confident errors. Finally,

we aggregate signals with a gate-by-gate cascade that mirrors Figure 1. Let $\bar{\mathbf{p}}$ be the averaged scores, $y^\star/r$ the consensus, S the stability, and Π pragmatic tags. Algorithm 1 lists the fusion procedure.

Filtering Policy

(1) Validity: If c is empty or $\bar{\mathbf{p}}$ missing, return `neutral`. (2) Stability: If $S < \tau_{\text{unstable}}$, return `neutral`. (3) Pragmatics-aware confidence: Let $y = \arg\max \bar{\mathbf{p}}$ and $p_{\max} = \max \bar{\mathbf{p}}$; if `Sarcasm` or `Negation` $\in \Pi$, subtract a penalty λ_{prag} from $p_{\max}$. (4) Consensus override under low confidence: If adjusted $p_{\max} < \tau_{\text{prob}}$ and $r \geq \tau_{\text{cons}}$, output $y^\star$; else `neutral`. (5) Prefer consensus: If $r \geq \tau_{\text{cons}}$, output $y^\star$. (6) Small-margin flip: When $y =$ `favor` and lead $= p_{\text{favor}} - \max(p_{\text{against}}, p_{\text{neutral}}) \leq \tau_{\text{flip}}$, flip to `against` if `Sarcasm` or `Negation` $\in \Pi$. (7) Tie-to-neutral: If the top-2 gap $< \epsilon$, output `neutral`. (8) Fallback: Otherwise output y.

Algorithm 1 Stability-Aware Fusion.

Require: Π, $\bar{\mathbf{p}}$, $y^\star$, r, S; thresholds $\tau_{\text{unstable}}, \tau_{\text{prob}}, \tau_{\text{cons}}, \tau_{\text{flip}}, \epsilon$; penalty λ_{prag}

if c empty or $\bar{\mathbf{p}}$ missing **then return** `neutral`
if $S < \tau_{\text{unstable}}$ **then return** `neutral`
$y \leftarrow \arg\max_i p_i$, $p_{\max} \leftarrow \max_i p_i$ ▷ $\bar{\mathbf{p}} = (p_{\text{favor}}, p_{\text{against}}, p_{\text{neutral}})$
if `Sarcasm`$\in \Pi$ **or** `Negation`$\in \Pi$ **then**
    $\tilde{p}_{\max} \leftarrow \max(0,\ p_{\max} - \lambda_{\text{prag}})$
else
    $\tilde{p}_{\max} \leftarrow p_{\max}$
end if
if $\tilde{p}_{\max} < \tau_{\text{prob}}$ **then**
    if $r \geq \tau_{\text{cons}}$ **then**
        return $y^\star$
    else
        return `neutral`
    end if
end if
if $r \geq \tau_{\text{cons}}$ **then**
    return $y^\star$
end if
if $y =$ `favor` **then**
    lead $\leftarrow p_{\text{favor}} - \max(p_{\text{against}}, p_{\text{neutral}})$
    if lead $\leq \tau_{\text{flip}}$ **and** (`Sarcasm`$\in \Pi$ **or** `Negation`$\in \Pi$) **then**
        return `against`
    end if
end if
Let $p_{(1)} \geq p_{(2)}$ be the top-2 of $\bar{\mathbf{p}}$
if $|p_{(1)} - p_{(2)}| < \epsilon$ **then**
    return `neutral`
end if
return y

4. Experiments

4.1. Experimental Data

To evaluate the effectiveness of our approach, we run thorough experiments on two datasets: SemEval-2016 Task 6 (SEM16) [2] and COVID-19-Stance (COVID19) [31].

- SemEval-2016 (SEM16) [2]: The SEM16 dataset contains 4870 tweets, each targeting various subjects and annotated with one of three stance labels: "favor", "against", or "neutral". SEM16 provides six targets (Donald Trump (DT), Hillary Clinton (HC), Feminist Movement (FM), Legalization of Abortion (LA), Atheism (AT), and Climate Change (CC)). Following [25] for zero-shot evaluation, we exclude Atheism (AT) and

Climate Change (CC) and use only the official test split for fair comparison with prior zero-shot settings. Per-target class counts for the four retained targets are reported in Table 1.

- COVID-19-Stance (COVID-19) [31]: We also use the COVID-19 stance dataset, which assesses public attitudes toward pandemic-related policies and figures across four targets: *Wearing a Face Mask* (WA), *Keeping Schools Closed* (SC), *Anthony S. Fauci, M.D.* (AF), and *Stay at Home Orders* (SH). Each tweet is labeled with *Favor*/*Against*/*Neutral*. As with SEM16, we report results using the test split only; class distributions are shown in Table 1.

Table 1. Statistics of the datasets used in our experiments.

Dataset	Target	Favor	Against	Neutral	Total
SEM16	DT	148	299	260	707
	HC	163	565	251	979
	FM	268	511	170	949
	LA	167	544	222	933
COVID-19	AF	492	610	762	1864
	SH	615	250	325	1190
	WA	693	190	668	1551
	SC	400	782	346	1528

Both datasets were chosen as they are widely adopted benchmarks for stance detection and provide diverse, topic-specific challenges. SEM16 offers classic political and social targets that test generalization across ideological domains, while COVID-19-Stance introduces a contemporary and highly pragmatic context involving public health discourse. We acknowledge that both datasets are relatively small in size (a few thousand tweets each), which may limit the statistical power of comparisons. To mitigate this, all evaluations are conducted in a strict zero-shot setting using the official test splits only, ensuring comparability with prior studies. Furthermore, our stability-aware fusion mechanism and counterfactual probing explicitly reduce the sensitivity of results to sampling variance, partially alleviating dataset-size constraints.

4.2. Evaluation Metrics

We adopt the Macro-F1 score F_{avg} computed as the average of the F1 scores of the *Favor* and *Against* categories (the *Neutral* class is excluded from the average), following standard practice in stance detection [2,7,26]. Macro-F1 equally weights each class and is less affected by label imbalance, which is critical for social media datasets where "neutral" or "against" instances often dominate. In addition, we report per-target Macro-F1 averages to ensure comparability with prior zero-shot studies.Let P_y and R_y denote precision and recall for class $y \in \{\text{favor}, \text{against}\}$ The per-class F1 and the macro average are defined as

$$F_{\text{favor}} = \frac{2 \cdot P_{\text{favor}} \cdot R_{\text{favor}}}{P_{\text{favor}} + R_{\text{favor}}}, \tag{1}$$

$$F_{\text{against}} = \frac{2 \cdot P_{\text{against}} \cdot R_{\text{against}}}{P_{\text{against}} + R_{\text{against}}}, \tag{2}$$

$$F_{\text{avg}} = \frac{F_{\text{favor}} + F_{\text{against}}}{2}. \tag{3}$$

4.3. Baseline Methods

To ensure a comprehensive evaluation, we compare PAMR with a broad range of existing stance detection approaches, which can be grouped into traditional DNNs, fine-tuning-based methods, and LLM-based frameworks.

(1) TraditionalDNNs.

- **BiLSTM and Bicond [32]**: These two approaches utilize separate BiLSTM encoders, where one captures sentence-level semantics and the other encodes the given target, thereby enabling the model to jointly represent stance-related information.
- **CrossNet [33]**: This model leverages BiLSTM architectures to encode both the input text and its corresponding target, while introducing a target-specific attention mechanism before classification, which enhances the model's ability to generalize across unseen targets.
- **TPDG [34]**: This method automatically identifies stance-bearing words and distinguishes target-dependent from target-independent terms, adjusting them adaptively to better capture the relationship between text and target.
- **TOAD [35]**: To improve generalization in zero-shot scenarios, TOAD adopts an adversarial learning strategy that allows the model to resist overfitting to specific targets while transferring stance knowledge.

(2) Fine-tuning methods.

- **TGA-Net [36]**: This approach establishes associations between training and evaluation topics in an unsupervised way, using BERT as the encoder and fully connected layers for classification, thereby linking topic domains without annotated supervision.
- **Bert-Joint [37]**: It combines bidirectional encoder representations from transformers that have been pre-trained on large-scale unlabeled corpora, producing dense contextual embeddings for both tokens and full sentences.
- **Bert-GCN [38]**: This method enhances stance detection with common-sense knowledge by integrating both structural and semantic graph relations, which makes it more effective in generalizing to zero and few-shot target scenarios.
- **JointCL [24]**: It unifies stance-oriented contrastive learning with target-aware prototypical graph contrastive learning, allowing the model to transfer stance-relevant features learned from seen topics to unseen targets.
- **TarBK [39]**: By leveraging Wikipedia-derived background knowledge, TarBK reduces the semantic gap between training and evaluation targets, thereby improving the reasoning capability of stance classifiers.
- **PT-HCL [7]**: This contrastive learning approach utilizes both semantic and sentiment features to improve cross-domain stance transferability, enabling robust generalization beyond source data. Its combination of semantics and sentiment proves critical for disambiguating stance from emotion.
- **KEPrompt [40]**: This method proposes an automatic verbalizer to generate label words dynamically, while simultaneously injecting external background knowledge to guide stance recognition. It reduces reliance on manually designed verbalizers and improves flexibility.

(3) LLM-based methods.

- **COLA [14]**: This approach employs a three-stage framework where different LLM roles are orchestrated for multidimensional text understanding and reasoning, resulting in state-of-the-art zero-shot stance performance. It demonstrates the effectiveness of task decomposition within LLM pipelines.

- **Ts-CoT [41]**: It introduces a chain-of-thought prompting mechanism for stance detection with LLMs, upgrading the base model to GPT-3.5 in order to take advantage of improved reasoning capacity. The CoT design encourages step-by-step reasoning rather than direct label prediction.
- **EDDA [27]**: This method exploits LLMs to automatically generate rationales and substitute stance-bearing expressions, thereby increasing semantic relevance and expression diversity for stance detection. By focusing on rationales, EDDA improves both performance and interpretability.
- **FOLAR [42]**: A reasoning framework that augments stance detection with factual knowledge and chain-of-thought logical reasoning, aiming to improve interpretability and robustness in zero-shot settings.
- **LogiMDF [25]**: A logic-augmented multi-decision fusion framework that extracts first-order logic rules from multiple LLMs, constructs a logical fusion schema, and employs a multi-view hypergraph neural network to integrate diverse reasoning processes for consistent and accurate stance detection.

4.4. Implementation Details

We implement PAMR using `GPT-3.5-turbo` accessed via the OpenAI API. For Linguistic Parse and CVS, the maximum generation length is set to 256 and 512 tokens, respectively, with temperature 0.3. For NLI-based estimation, we run the model three times with temperatures $\{0.3, 0.5, 0.8\}$ and aggregate results by majority vote and averaged probabilities. Fusion thresholds are fixed across datasets: $\tau_{\text{unstable}} = 0.40$, $\tau_{\text{prob}} = 0.45$, $\tau_{\text{cons}} = 0.60$, $\tau_{\text{flip}} = 0.15$, $\epsilon = 0.02$, and pragmatic penalty $\lambda_{\text{prag}} = 0.05$. All evaluations are conducted in a strict zero-shot setting using the official test splits, and performance is reported in terms of macro-F1 over `favor` and `against`.

5. Overall Performance

5.1. Analysis of Main Results

Table 2 reports the results on the SEM16 dataset. Early neural baselines such as BiLSTM, Bicond, and CrossNet achieve relatively low performance, with average scores around 34–38. With the introduction of pre-trained models, methods like JointCL, PT-HCL, and NPS4SD raise the average performance to around 52–55. More recent LLM-based approaches, such as COLA, FOLAR, and LogiMDF, achieve competitive results on specific targets (e.g., COLA and FOLAR both exceed 81 on HC). However, their performance tends to fluctuate across different targets. In contrast, PAMR achieves strong and consistent results across all targets, with best scores on FM (75.9) and LA (71.8), and a balanced overall average of 71.9, surpassing all baselines. This demonstrates the effectiveness of pragmatic-aware reasoning and stability fusion in achieving robust zero-shot stance detection. Here, Avg denotes the arithmetic mean over all targets.

Table 3 shows the results on the COVID-19 dataset. Traditional baselines (CrossNet, BERT, TPDG) perform poorly, with average scores between 40 and 45. Stronger models like TOAD and JointCL improve performance, while LogiMDF achieves the best results on AF (70.4) and WA (75.4), and FOLAR performs strongly on WA (73.1). Nevertheless, PAMR consistently achieves competitive or best performance on nearly all targets, obtaining the top scores on SC (72.0), SH (72.2), and WA (78.8). Its overall average reaches 73.0, outperforming all comparison methods. This indicates that PAMR is better able to handle pragmatic confounds such as sarcasm and negation in COVID-19 tweets, yielding more stable and reliable stance predictions. Again, Avg is the mean over all targets in the dataset.

Table 2. Results on the SEM16 dataset. F_{avg} is reported for each target. The Avg is the arithmetic mean over all listed targets. The best results are in bold. ‡ indicates the first-best result.

Method	HC	FM	LA	DT	Avg
BiLSTM	31.6	40.3	33.6	30.8	34.1
Bicond	32.7	40.6	34.4	30.5	34.6
CrossNet	38.3	41.7	38.5	35.6	38.5
TPDG	50.9	53.6	46.5	47.3	49.6
TOAD	51.2	54.1	46.2	49.5	50.3
TGA-Net	49.3	46.6	45.2	40.7	45.5
Bert-Joint	50.1	42.1	44.8	41.0	44.5
Bert-GCN	50.0	44.3	44.2	42.3	45.2
JointCL	54.4	54.0	50.0	50.5	52.2
TarBK	55.1	53.8	48.7	50.8	52.1
PT-HCL	54.5	54.6	50.9	50.1	52.5
KEPROMPT	57.0	53.6	53.0	41.8	51.3
NPS4SD	60.1	56.7	51.0	51.4	54.8
COLA	**81.7**	63.4	71.0	68.5	71.2
Ts-CoT$_{GPT}$	78.9	68.3	62.3	68.6	69.5
EDDA	77.4	69.7	62.7	**69.8**	69.9
FOLAR	81.9	71.2	69.9	–	–
LogiMDF	75.1	67.9	68.0	67.6	69.7
PAMR (Ours)	73.7	**75.9** ‡	**71.8** ‡	66.1	**71.9**

Table 3. Results on the COVID-19 dataset. We report per-target Macro-F1 and the overall Avg, which is the mean across all targets. The best numbers are in bold. ‡ indicates the first-best result.

Method	AF	SC	SH	WA	Avg
CrossNet	41.3	40.0	40.4	38.2	40.0
BERT	47.3	45.0	39.9	44.3	44.1
TPDG	46.0	51.5	37.2	48.0	45.7
TOAD	53.0	68.3	62.9	41.1	56.3
JointCL	57.6	49.3	43.5	63.1	53.4
Ts-CoT	69.2	43.5	66.5	57.8	59.3
COLA	65.7	46.6	53.5	73.9	59.9
FOLAR	69.5	67.2	65.4	73.1	68.8
LogiMDF	**70.4**	68.8	64.9	75.4	69.9
PAMR	68.6	**72.0** ‡	**72.2** ‡	**78.8** ‡	**73.0**

To further validate these improvements, we conduct paired *t*-tests between PAMR and the strongest baseline across all targets on both datasets. The results show that PAMR's performance gains are statistically significant ($p < 0.05$), confirming that the observed advantages are unlikely due to random variation.

5.2. Ablation Study

To better understand the contributions of different components in PAMR, we conduct ablation experiments by removing the Linguistic Parser (w/o LP) and the Counterfactual and View-Switching module (w/o CVS). The results are shown in Figure 2.

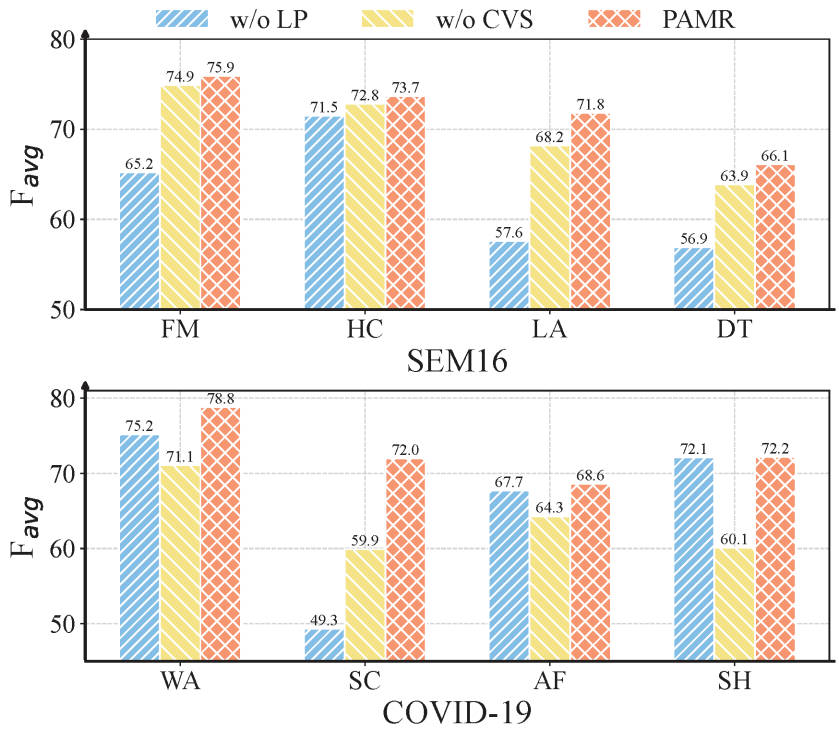


Figure 2. Ablation study of PAMR on SEM16 and COVID-19.

We observe that removing either component consistently degrades performance across both COVID-19 and SemEval-2016 datasets. Specifically, eliminating the Linguistic Parser leads to a substantial drop, e.g., from 72.0 to 49.3 on SC and from 66.1 to 56.9 on DT, confirming the importance of extracting canonical claims and pragmatic cues for stance reasoning. Similarly, removing the Counterfactual and View-Switching module also results in performance declines, particularly on WA (from 78.8 to 71.1) and FM (from 75.9 to 74.9), highlighting the necessity of stability probing to mitigate pragmatic confounds and prevent over-confident errors.

Overall, the full PAMR framework consistently outperforms its ablated variants, demonstrating that both pragmatic parsing and counterfactual stability probing are complementary and essential for robust zero-shot stance detection.

5.3. Case Study

To further illustrate the behavior of PAMR, we analyze representative examples from the evaluation set, as shown in Table 4.

Case 1. In this example, the canonical claim and the original tweet exhibit strong semantic alignment. The NLI-based estimation assigns a dominant probability to `favor` ($p = 0.65$), clearly surpassing both `against` and `neutral`. The CVS stability score reaches $S = 0.83$, well above the instability threshold, indicating robust consistency under counterfactual and perspective shifts. Moreover, majority voting yields complete unanimity (3/3 runs), reinforcing the reliability of the prediction. Since no pragmatic markers (e.g., sarcasm or negation) are present, the system directly outputs `favor`, which matches the gold annotation. This case shows that, under standard conditions, PAMR can capture entailment relations both stably and accurately.

Table 4. Case studies on **SemEval-2016** Trump target showing intermediate outputs.

Tweet	Claim	Pragmatic Tags	NLI Scores	Stability	Prediction
People are saying that kids will not have a safe place to go to if the schools are closed. Large gatherings are not safe with coronavirus. The coronavirus is not safe.	Keeping schools closed may leave kids without a safe place to go, but large gatherings are unsafe due to coronavirus.	[Quotation]	{favor: 0.65, against: 0.15, none: 0.20}	0.833	Favor
"2 years ago Hillary never answered whether she used private email. Liberal media passed on reporting." #equality	Questioning Hillary's email while excusing others reflects the sexist double standards women in politics face.	[Sarcasm, Quotation]	{favor: 0.50, against: 0.46, none: 0.04}	0.666	Against

Case 2. Here, the canonical claim and initial NLI probabilities suggest an approval stance (`favor`) toward the target, with the aggregated vote leaning in the same direction. However, the CVS stability score is only $S = 0.67$, indicating borderline robustness. Importantly, pragmatic markers such as `Sarcasm` are detected. Within the fusion stage, these cues trigger a polarity flip, shifting the final decision from `favor` to `against`, which aligns with the gold annotation. This case demonstrates PAMR's ability to exploit pragmatic signals to correct initial misclassifications in sarcastic contexts, thereby disentangling stance polarity from sentiment polarity.

6. Discussion

Our results highlight key insights into zero-shot stance detection. PAMR consistently outperforms baselines on both SEM16 and COVID-19 datasets, demonstrating that explicit modeling of pragmatics and stability improves generalization. Unlike prior models that conflate sentiment with stance, PAMR leverages pragmatic cues like sarcasm and negation to reduce polarity errors. Its modular design—with components for claim extraction, counterfactual probing, and fusion—enables interpretable outputs and fine-grained analysis. While PAMR currently relies on GPT-3.5, all reasoning steps are prompt-based and generalizable, making the framework compatible with future open-source or symbolic alternatives. Fusion strategies further enhance robustness under figurative or domain-specific inputs. Future work may explore lightweight replacements to improve accessibility and stability.

7. Conclusions

In this paper, we introduced PAMR, a pragmatic-aware multi-agent reasoning framework for zero-shot stance detection on social media. The framework enables explicit and interpretable reasoning by integrating linguistic, inferential, and counterfactual analyses without requiring target-specific training. Experiments on the SemEval-2016 and COVID-19-Stance datasets show that PAMR achieves stable improvements over strong zero-shot baselines, confirming the value of incorporating pragmatic cues and stability probing. While our results demonstrate promising robustness and interpretability, the model has been evaluated only on English social media datasets; its applicability to other domains or high-stakes contexts should be considered exploratory. Future work will extend

PAMR to multilingual and multi-modal settings, investigate domain adaptation under distribution shifts, and explore hybrid integration with symbolic reasoning to further enhance stability and transparency in zero-shot stance detection. Beyond these directions, PAMR offers a flexible foundation for future research on pragmatic reasoning and stance analysis. Its modular components can be adapted to evaluate or enhance other stance models, while its interpretable outputs provide useful tools for analyzing how pragmatic cues shape stance, supporting broader advancements in social media understandings.

Author Contributions: Conceptualization, Z.X. and B.Z.; Methodology, Z.X. and F.N.; Investigation, G.D.; Writing—original draft, Z.X. and F.N.; Writing—review and editing, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Natural Science Foundation of Top Talent of SZTU (grant no. GDRC202320) and the Research Promotion Project of Key Construction Discipline in Guangdong Province (2022ZDJS112).

Data Availability Statement: No new data were created in this study. The datasets analyzed are publicly available benchmark datasets (SemEval-2016 and COVID-19), which can be obtained from their respective sources. Processed data and code are available from the corresponding author upon reasonable request.

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

References

1. Küçük, D.; Can, F. Stance detection: A survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 12. [CrossRef]
2. Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; Cherry, C. Semeval-2016 task 6: Detecting stance in tweets. In Proceedings of the 10th International Workshop On Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 31–41.
3. Addawood, A.; Schneider, J.; Bashir, M. Stance classification of twitter debates: The encryption debate as a use case. In Proceedings of the 8th International Conference on Social Media & Society, Toronto, ON, Canada, 28–30 July 2017; pp. 1–10.
4. Sun, Q.; Wang, Z.; Zhu, Q.; Zhou, G. Stance detection with hierarchical attention network. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 21–24 August 2018; pp. 2399–2409.
5. Li, Y.; Caragea, C. Target-Aware Data Augmentation for Stance Detection. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 1850–1860.
6. Xu, C.; Paris, C.; Nepal, S.; Sparks, R. Cross-target stance classification with self-attention networks. *arXiv* **2018**, arXiv:1805.06593.
7. Liang, B.; Chen, Z.; Gui, L.; He, Y.; Yang, M.; Xu, R. Zero-Shot Stance Detection via Contrastive Learning. In Proceedings of the ACM Web Conference 2022, Virtual, 25–29 April 2022; pp. 2738–2747.
8. Hong, G.N.S.Y.; Gauch, S. Sarcasm Detection as a Catalyst: Improving Stance Detection with Cross-Target Capabilities. *arXiv* **2025**, arXiv:2503.03787. [CrossRef]
9. Maynard, D.; Greenwood, M. Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*; Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., Eds.; European Language Resources Association (ELRA): Reykjavik, Iceland, 2014; pp. 4238–4243.
10. Du, J.; Xu, R.; He, Y.; Gui, L. Stance classification with target-specific neural attention networks. In Proceedings of the International Joint Conferences on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017.
11. Wei, P.; Lin, J.; Mao, W. Multi-target stance detection via a dynamic memory-augmented network. In Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 1229–1232.
12. Li, Y.; Sosea, T.; Sawant, A.; Nair, A.J.; Inkpen, D.; Caragea, C. P-stance: A large dataset for stance detection in political domain. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; pp. 2355–2365.
13. Conforti, C.; Berndt, J.; Pilehvar, M.T.; Giannitsarou, C.; Toxvaerd, F.; Collier, N. Synthetic Examples Improve Cross-Target Generalization: A Study on Stance Detection on a Twitter corpus. In Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EACL 2021, Online, 19 April 2021; pp. 181–187.

14. Lan, X.; Gao, C.; Jin, D.; Li, Y. Stance detection with collaborative role-infused llm-based agents. In Proceedings of the International AAAI Conference on Web and Social Media, Buffalo, NY, USA, 3–6 June 2024; Volume 18, pp. 891–903.
15. Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. *arXiv* **2024**, arXiv:2402.01680. [CrossRef]
16. Tran, K.T.; Dao, D.; Nguyen, M.D.; Pham, Q.V.; O'Sullivan, B.; Nguyen, H.D. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. *arXiv* **2025**, arXiv:2501.06322. [CrossRef]
17. Zarrella, G.; Marsh, A. MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 458–463.
18. Sobhani, P.; Inkpen, D.; Zhu, X. A dataset for multi-target stance detection. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, 3–7 April 2017; pp. 551–557.
19. He, Z.; Mokhberian, N.; Lerman, K. Infusing Wikipedia Knowledge to Enhance Stance Detection. *arXiv* **2022**, arXiv:2204.03839.
20. Wang, S.; Pan, L. Target-Adaptive Consistency Enhanced Prompt-Tuning for Multi-Domain Stance Detection. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia, 20–25 May 2024; pp. 15585–15594.
21. Nguyen, D.Q.; Vu, T.; Nguyen, A.T. BERTweet: A Pre-trained Language Model for English Tweets. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 9–14. [CrossRef]
22. Müller, M.; Salathé, M.; Kummervold, P.E. COVID-Twitter-BERT: A Natural Language Processing Model to Analyse COVID-19 Content on Twitter. *arXiv* **2020**, arXiv:2005.07503. [CrossRef] [PubMed]
23. Liang, B.; Li, A.; Zhao, J.; Gui, L.; Yang, M.; Yu, Y.; Wong, K.F.; Xu, R. Multi-modal Stance Detection: New Datasets and Model. *arXiv* **2024**, arXiv:2402.14298. [CrossRef]
24. Liang, B.; Zhu, Q.; Li, X.; Yang, M.; Gui, L.; He, Y.; Xu, R. Jointcl: A joint contrastive learning framework for zero-shot stance detection. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Volume 1, pp. 81–91.
25. Zhang, B.; Ma, J.; Fu, X.; Dai, G. Logic Augmented Multi-Decision Fusion Framework for Stance Detection on Social Media. *Inf. Fusion* **2025**, *122*, 103214. [CrossRef]
26. Li, A.; Liang, B.; Zhao, J.; Zhang, B.; Yang, M.; Xu, R. Stance Detection on Social Media with Background Knowledge. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 15703–15717.
27. Ding, D.; Dong, L.; Huang, Z.; Xu, G.; Huang, X.; Liu, B.; Jing, L.; Zhang, B. EDDA: An Encoder-Decoder Data Augmentation Framework for Zero-Shot Stance Detection. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia, 20–25 May 2024; pp. 5484–5494.
28. Fei, H.; Li, B.; Liu, Q.; Bing, L.; Li, F.; Chua, T.S. Reasoning Implicit Sentiment with Chain-of-Thought Prompting. *arXiv* **2023**, arXiv:2305.11255.
29. Ling, Z.; Fang, Y.; Li, X.; Huang, Z.; Lee, M.; Memisevic, R.; Su, H. Deductive Verification of Chain-of-Thought Reasoning. *arXiv* **2023**, arXiv:2306.03872.
30. Cai, Z.; Chang, B.; Han, W. Human-in-the-Loop through Chain-of-Thought. *arXiv* **2023**, arXiv:2306.07932.
31. Glandt, K.; Khanal, S.; Li, Y.; Caragea, D.; Caragea, C. Stance detection in COVID-19 tweets. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Long Papers), Online, 1–6 August 2021; Volume 1.
32. Augenstein, I.; Rocktaeschel, T.; Vlachos, A.; Bontcheva, K. Stance Detection with Bidirectional Conditional Encoding. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Sheffield, Austin, TX, USA, 1–5 November 2016.
33. Du, J.; Xu, R.; He, Y.; Gui, L. Stance Classification with Target-specific Neural Attention. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, Melbourne, Australia, 19–25 August 2017; pp. 3988–3994. [CrossRef]
34. Liang, B.; Fu, Y.; Gui, L.; Yang, M.; Du, J.; He, Y.; Xu, R. Target-Adaptive Graph for Cross-Target Stance Detection. In Proceedings of the WWW '21: The Web Conference 2021, Virtual Event/Ljubljana, Slovenia, 19–23 April 2021; pp. 3453–3464.
35. Allaway, E.; Srikanth, M.; Mckeown, K. Adversarial Learning for Zero-Shot Stance Detection on Social Media. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 4756–4767.
36. Allaway, E.; Mckeown, K. Zero-Shot Stance Detection: A Dataset and Model using Generalized Topic Representations. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 8913–8931.

37. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
38. Liu, R.; Lin, Z.; Tan, Y.; Wang, W. Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021, pp. 3152–3157.
39. Zhu, Q.; Liang, B.; Sun, J.; Du, J.; Zhou, L.; Xu, R. Enhancing Zero-Shot Stance Detection via Targeted Background Knowledge. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 2070–2075.
40. Huang, H.; Zhang, B.; Li, Y.; Zhang, B.; Sun, Y.; Luo, C.; Peng, C. Knowledge-Enhanced Prompt-Tuning for Stance Detection. *ACM Trans. Asian-Low-Resour. Lang. Inf. Process.* **2023**, *22*, 159. [CrossRef]
41. Zhang, B.; Fu, X.; Ding, D.; Huang, H.; Li, Y.; Jing, L. Investigating Chain-of-thought with ChatGPT for Stance Detection on Social Media. *arXiv* **2023**, arXiv:2304.03087.
42. Dai, G.; Liao, J.; Zhao, S.; Fu, X.; Peng, X.; Huang, H.; Zhang, B. Large Language Model Enhanced Logic Tensor Network for Stance Detection. *Neural Netw.* **2025**, *183*, 106956. [CrossRef] [PubMed]

electronics

MDPI

Article

Enhancing the Sustained Capability of Continual Test-Time Adaptation with Dual Constraints

Yu Song, Pei Liu and Yunpeng Wu *

School of Computer Science and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China; ieysong@zzu.edu.cn (Y.S.); peiliu123@gs.zzu.edu.cn (P.L.)
* Correspondence: ieypwu@zzu.edu.cn

Abstract: Continuous Test-Time Adaptation aims to adapt a source model to continuously and dynamically changing target domains. However, previous studies focus on adapting to each target domain independently, treating them as isolated, while ignoring the interplay of interference and promotion between domains, which limits the model's sustained capability, often causing it to become trapped in local optima. This study highlights this critical issue and identifies two key factors that limit the model's sustained capability: (1) The update of parameters lacks constraints, where domain-sensitive parameters capture domain-specific knowledge, leading to unstable channel representations and interference from old domain knowledge and hindering the learning of domain-invariant knowledge. (2) The decision boundary lacks constraints, and distribution shifts, which carry significant domain-specific knowledge, cause features to become dispersed and prone to clustering near the decision boundary. This is particularly problematic during the early stages of domain shifts, where features are more likely to cross the boundary. To tackle the two challenges, we propose a Dual Constraints method: First, we constrain updates to domain-sensitive parameters by minimizing the representation changes in domain-sensitive channels, alleviating the interference among domain-specific knowledge and promoting the learning of domain-invariant knowledge. Second, we introduce a constrained virtual decision boundary, which forces features to move away from the original boundary, and with a virtual margin to prevent features from crossing the decision boundary due to domain-specific knowledge interference caused by domain shifts. Extensive benchmark experiments show our framework outperforms competing methods.

Keywords: continual test-time adaption; test-time adaption; domain adaption; domain generalization; continual learning

1. Introduction

Continual Test-Time Adaptation (CTTA) focuses on enhancing machine learning models' ability to adapt continuously in dynamic environments where input data distributions shift over time. This capability is crucial in various real-world scenarios. For instance, in autonomous driving [1–4], a vehicle may encounter changing conditions such as transitioning from daylight to nighttime or from sunny to rainy weather. To maintain high performance, models must adapt effectively to these evolving data distributions through Continual Test-Time Adaptation.

Numerous methods have been developed to tackle the challenges of Continual Test-Time Adaptation, including the use of teacher–student models [5–7], data augmentation techniques [5], semi-supervised learning [8], Low-Rank Learning [9], sample replay [10], and Masked Autoencoders [11].

Although previous studies have achieved significant success, a key limitation must be pointed out: These studies still follow the approach of Test-Time Adaptation (TTA) [12], treating the problem of Continual Test-Time Adaptation in isolation and focusing solely on adapting to each domain independently. This limits the model's sustained capability, similar to a greedy algorithm [13], causing the model's performance to become trapped in local optima [14] and failing to achieve the ideal global optimum [15]. This is because these methods overlook the essential nature of Continual Test-Time Adaptation: in this setting, domain adaptation is dynamic, ongoing, and continuously evolving, resulting in different domains either interfering with or promoting each other.

Therefore, addressing the following two challenges is crucial for enhancing the sustained capability of Continual Test-Time Adaptation: (1) How to effectively alleviate the mutual interference of domain-specific knowledge. (2) How to effectively learn domain-invariant knowledge across domains.

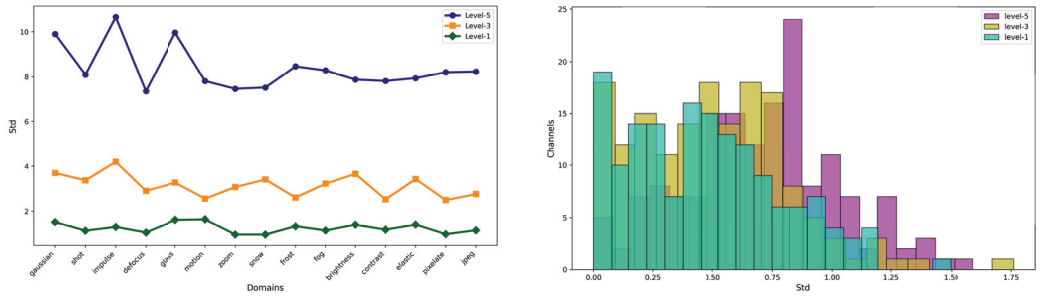
This study revisits the differences between Continual Test-Time Adaptation and Test-Time Adaptation, focusing on the long-neglected issue of sustained capability, and identifies two key factors that severely limit the model's sustained capability and prevent it from getting trapped in local optima:

1. Lack of constraints in parameter updates: The model contains a large number of domain-sensitive parameters, which tend to learn domain-specific knowledge. When the model encounters a new domain, the distribution difference between the old and new domains causes these domain-sensitive parameters to behave abnormally, leading to unstable channel representations that incorporate substantial domain-specific knowledge from previous domains, especially in the early stages of adapting to the new domain. This results in severe interference between domain-specific knowledge. More importantly, relying on domain-specific knowledge for classification significantly hinders the learning of domain-invariant knowledge.
2. Lack of constraints in decision boundary: Domain shifts carry a significant amount of domain-specific knowledge, causing the features generated by the model to spread out more. The lack of constraints makes these features prone to clustering near or crossing the decision boundary. Under the interference of domain-specific knowledge, features near the decision boundary are more likely to cross, particularly in the early stages of the new domain.

As shown in Figure 1, we analyze the channel representations under three different corruption levels (level-5, level-3, and level-1) of CIFAR10C (The method for calculating unstable channel representations is provided in Section 4.4.3). In Figure 1a, we observe that different levels of corruption have varying impacts on the stability of channel representations. The higher the level of corruption, the more unstable the channel representations become. The domain-sensitive parameters are highly sensitive to this domain-specific knowledge, resulting in anomalies and instability in the channel representations.

In Figure 1b, we observe that domain-sensitive channels are much more responsive to domain shifts than domain-robust channels. When encountering different domains, the representation of these channels tends to exhibit instability and are prone to significant variations. The cause of this abnormal behavior lies in the domain-sensitive parameters, which tend to overfit domain-specific knowledge such as background, lighting, and texture. When a domain shift occurs, this domain-specific knowledge undergoes drastic changes, leading to significant variations in the channel representations. The instability of channel representations and the presence of a large amount of domain-specific knowledge can cause significant interference in domain adaptation, especially in the early stages of domain shift. In contrast, domain-robust parameters focus on learning domain-invariant knowledge,

such as contours and shapes. This domain-invariant knowledge remains unaffected by domain shifts, resulting in more stable channel representations.



(**a**) (**b**)

Figure 1. An illustration of the relationship between channels and unstable representations: (**a**) The instability of channel representations increases with the level of corruption. This indicates that domain shifts impact the channel outputs, and the stronger the domain shift, the greater the instability of the channel outputs. (**b**) The instability of representations varies across different channels. Domain-robust channels exhibit stable representations with smaller variance, typically concentrated on the left, suggesting that these channels have learned more domain-invariant knowledge, making them resilient to data distribution shifts and less prone to changes. In contrast, domain-sensitive channels show unstable representations with larger variance, typically concentrated on the right, as they have learned domain-specific knowledge, making them vulnerable to data distribution shifts and more susceptible to changes.

Based on our observation in Figure 1, we can see that domain-specific channels exhibit instability and are prone to change. Therefore, we propose a parameter update constraint method that estimates the relationship between changes in channel representations and the loss increases caused by parameter updates, suppressing domain-sensitive parameters by minimizing the changes in domain-sensitive channel representations. This constraint not only effectively alleviates the interference caused by domain-specific knowledge but also promotes the learning of domain-invariant knowledge by reducing the sensitivity of the parameters. Additionally, we provide theoretical evidence that our method can effectively enhance the model's generalization ability and promote the learning of domain-invariant knowledge.

Furthermore, as shown in Figure 2, domain shifts carry substantial domain-specific knowledge, and when the model has not yet adapted to the new domain, features are more susceptible to the effects of the domain shift, making them more likely to cross the original decision boundary. We introduce a virtual decision boundary that constrains the features generated by the model to move away from the original decision boundary, preventing them from clustering near it. This constraint also creates a virtual margin between two decision boundaries. During domain shifts, this virtual margin provides sufficient buffering to prevent features from crossing the original decision boundary. The strongly constrained virtual decision boundary effectively mitigates the interference caused by domain-specific knowledge in the early stages of domain adaptation.

Overall, we propose a Dual Constraints method that combines channel-based parameter constraints and feature-based virtual decision boundary constraints, effectively addressing the two major challenges of domain knowledge interference and learning domain-invariant knowledge, thereby enhancing the model's sustained capability(More motivations and details will be elaborated in Section 3.1). Our contributions can be summarized as follows:

1. We propose a novel parameter constraint method that minimizes the representation changes in domain-sensitive channels, which, respectively, enhance and suppress the learning of domain-invariant and domain-specific knowledge. In addition, we theoretically prove that it can effectively enhance the model's generalization ability.
2. We introduce a strongly constrained virtual decision boundary that creates a virtual margin, forcing features away from the original decision boundary, effectively mitigating the problem of features crossing the boundary during domain shifts.
3. Dual Constraints enhance the model's sustained capability and achieve excellent performance, surpassing all existing state-of-the-art methods.

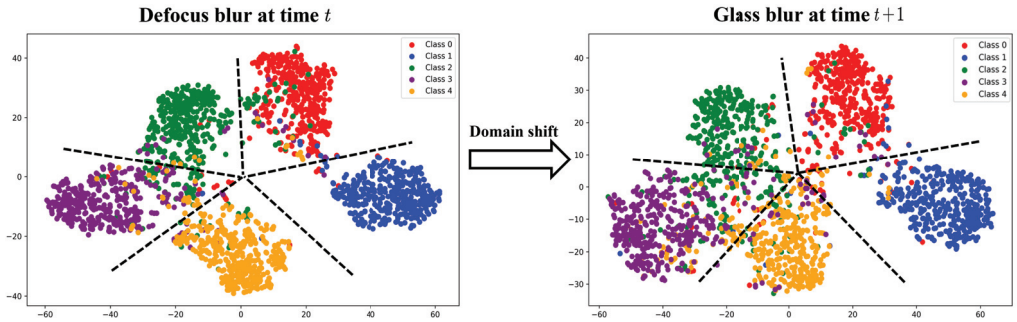


Figure 2. Decision boundary lacks constraints. Consider a domain shift occurring between two adjacent time points, t and $t+1$. Features that are clustered near the decision boundary at time t are subjected to a stronger domain shift, causing them to cross the decision boundary and manifest as features at time $t+1$.

2. Related Work

2.1. Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) [16] assumes that there is a domain shift between the source domain and the target domain, with labeled data available in the source domain but no labels in the target domain [17]. The goal is for the model to perform well on the unlabeled target domain data. UDA methods often use distribution distance metrics to align the feature distributions between the source and target domains during training. For example, Maximum Mean Discrepancy (MMD) [18–20] is a statistical test that assesses whether two distributions are equal based on observed samples from those distributions. Adversarial training [21–23] is another common approach, which involves aligning distributions using two adversarial roles: a domain classifier and a feature generator. Unlike traditional Unsupervised Domain Adaptation methods, Test-Time Adaptation (TTA) aims to adapt a model trained on a source domain to a new target domain without accessing the original source data during inference, and many methods have been proposed to solve TTA, such as TENT [12] and SHOT [24]. TENT [12] updates the trainable batch normalization parameters from a pretrained model by minimizing the entropy of the model's predictions during testing. SHOT [24] combines entropy minimization and diversity regularization with label smoothing techniques to train a general feature extractor from the pretrained source model.

2.2. Domain Generalization

Domain generalization (DG) aims to extract knowledge from the source domain that can generalize well to unseen target domains. Many methods learn domain-invariant representations by aligning the distributions of the target and source domains. These methods include adversarial learning [25], causal learning [26], and meta-learning [27]. Another approach is to enhance the model's generalization capability by generating more source domain data, specifically by augmenting the diversity of the source data through

data augmentation [28]. These data augmentation techniques primarily involve style transfer [28,29] and pixel-level augmentation [30]. Although these methods have demonstrated promising results, they may still learn excessive domain-specific features, as they rely on implicit assumptions to remove domain-specific characteristics via image-level augmentation or model-level constraints. Some studies have pointed out that Convolutional Neural Networks (CNNs) tend to classify objects based on local texture features that contain domain-specific characteristics [31–33]. To address this, they propose using penalty loss functions to suppress the model from learning local features such as texture, background, and lighting. This penalty encourages the model to rely on global features for classification. In contrast to local features, global features, including shape and contours, remain more stable and less prone to changes across domains. Therefore, this penalty on local features forces the model to learn domain-invariant properties, thereby enhancing its generalization ability.

2.3. Continual Learning

Continual learning (CL) aims to enable models to learn new tasks from a continuously evolving data stream while retaining previously acquired knowledge. One of the main challenges is catastrophic forgetting, where models tend to forget prior knowledge when learning new tasks. To address this issue, several approaches have been proposed, including regularization methods [34], which protect important weights from excessive updates, architecture expansion methods [35], which adapt the model by expanding its structure to accommodate new tasks, and memory replay methods [36–39], which store and replay data from previous tasks to mitigate forgetting. Despite significant progress, continual learning still faces challenges related to computational and memory overhead and maintaining strong generalization performance under non-stationary data distributions.

2.4. Continual Test-Time Adaptation

Unlike traditional Test-Time Adaptation (TTA) [40–42], which assumes a fixed target domain and works in a source-free online manner, CTTA accounts for the dynamic nature of real-world data distributions. This means CTTA requires models to adapt online and source-free across evolving domains. CoTTA [5] is a leading approach in continual domain learning, using average teachers and data augmentation for reliable pseudo-labels and robust model updates. DSS [8], inspired by semi-supervised learning, employs FreeMatch [43] to generate label thresholds for filtering pseudo-labels. RoTTA [44] stabilizes Batch Normalization updates to mitigate batch size variations during adaptation. RMT [6] addresses asymmetry in cross-entropy in teacher–student models, proposing symmetric cross-entropy for better gradients. Reshaping [10] addresses the catastrophic forgetting problem using sample replay. MAE [11] leverages mask autoencoders to learn domain-invariant knowledge. Unlike previous work, this study focuses on the sustainability of continuous domain adaptation, aiming to find the global optimum rather than settling for local optima.

3. Method

3.1. Problem Setting

Given a pretrained model M^0 with initial parameters θ_0, originally trained on source domain data (X^S, Y^S), the objective of CTTA is to iteratively adapt model M to a sequence of target domain datasets. During this process, the source domain data (X^S, Y^S) is not accessible, and the target domain datasets $\{X_0^T, X_1^T, \ldots, X_n^T\}$ are unlabeled. At each time step t, the parameters θ_t are updated to θ_{t+1} to better align the model M^{t+1} with the current target domain X_t^T. For the sake of clarity, we define the model output as:

$\hat{y} = M(x) = G(Z) = G_\ell(F_\ell(x))$. $F(x)$ is the feature extractor, and the feature map at the ℓ^{th} layer is $Z_\ell = \{Z_{\ell,1}, Z_{\ell,2}, \ldots, Z_{\ell,C_\ell}\} = F_{\ell-1}(Z_{\ell-1})$, where $Z_\ell \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}$. $G(\cdot)$ represents the classifier, and $W = \{W_1, W_2 \ldots W_C\}$ denotes the weights of the classifier.

3.2. Domain-Sensitive Parameter Suppression

3.2.1. Motivation of Domain-Sensitive Parameter Suppression

Existing methods typically use the following objective function to optimize model parameters for domain adaptation during testing:

$$\min_{M^{t+1}} \mathcal{L}(M^{t+1}(X_t^T), \hat{Y}_t^T) \tag{1}$$

where at time t, the target domain data X_t^T is encountered, and the model parameters are updated based on the loss function $\mathcal{L}(\cdot)$ to form the new model M^{t+1}. It is important to note that before any parameter update at time t, $M^{t+1} = M^t$.

Existing methods naively use Equation (1) as the objective function, leading to an unconstrained parameter update process. A large amount of domain-sensitive parameters are highly sensitive to domain-specific knowledge and change rapidly to fit such knowledge [31,45], such as background, lighting, and texture. Although existing methods introduce EMA [46], where $M^t = \beta M^{t-1} + (1-\beta)M^t$, and set $\beta = 0.999$ to alleviate performance degradation caused by rapid parameter changes, this parameter weighting approach has several issues: historical parameters contain a large amount of knowledge from past domains, which interferes with the current domain adaptation process, and hyperparameters cannot be adjusted adaptively. Moreover, this method is often used as a performance-boosting technique: people know that using it improves performance, but not using it worsens performance, without exploring the essence of the problem.

3.2.2. Implementation of Domain-Sensitive Parameter Suppression

To tackle this challenge, we propose a novel parameter constraint method that minimizes the representation changes in domain-sensitive channels, effectively mitigating the rapid updates and overfitting of domain-sensitive parameters to domain-specific knowledge.

Now, we consider two consecutive models, M^t and M^{t-1}. Theoretically, the parameter difference between two adjacent models is very small, but CTTA is a continuous process where the model continuously adapts to the target domain, which will eventually result in a huge cumulative effect. To analyze the feature maps change caused by the model parameter changes between adjacent moments, let Z'_ℓ and Z_ℓ are the feature maps generated by the $\ell - 1^{th}$ layer of $M^t(\cdot)$ and $M^{t-1}(\cdot)$, respectively. We can compute the loss given by Z'_ℓ via the first-order Taylor approximation as follows:

$$\begin{aligned} \mathcal{L}(G_\ell(Z'_\ell), \hat{y}) \approx\ & \mathcal{L}(G_\ell(Z_\ell), \hat{y}) \\ & + \sum_{c=1}^{C_\ell} \left\langle \nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y}), Z'_{\ell,c} - Z_{\ell,c} \right\rangle_F \end{aligned} \tag{2}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product, i.e., $\langle A, B \rangle_F = \mathrm{tr}(A^\top B)$, and $\nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y})$ is the gradient in the backpropagation of $M^{t-1}(\cdot)$. Using Equation (2), we define the loss increment $\triangle \mathcal{L}(Z'_{\ell,c})$ for the c^{th} channel at the ℓ^{th} layer caused by parameter changes as follows:

$$\triangle \mathcal{L}(Z'_{\ell,c}) := \left\langle \nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y}), Z'_{\ell,c} - Z_{\ell,c} \right\rangle_F \tag{3}$$

Thus, we aim to minimize the loss change caused by parameter updates between two adjacent models at consecutive moments, as follows:

$$\min_{M^{t+1}} \sum_{\ell=1}^{L} \sum_{c=1}^{C_\ell} \mathbb{E}_{(x)\sim X_t^T} \left[\triangle \mathcal{L}(Z'_{\ell,c}) \right]^2 \tag{4}$$

We need to update $M_t(\cdot)$ to $M_{t+1}(\cdot)$ at time t using data X_t^T. If we minimize Equation (4) using standard stochastic gradient descent, in addition to calculating the gradients of the feature maps produced by $M^t(\cdot)$ with respect to the samples, we also need to calculate and store the gradients of the feature maps produced by $M^{t-1}(\cdot)$ with respect to the samples, which significantly increases computational cost or memory overhead.

Using the Cauchy-Schwarz inequality, we derive the upper bound for the objective function of Equation (4) as follows:

$$\begin{aligned} &\mathbb{E}\Big[\triangle \mathcal{L}(Z'_{\ell,c})\Big] \\ &= \mathbb{E}\Big[\langle \nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y}), Z'_{\ell,c} - Z_{\ell,c} \rangle_F\Big] \\ &\leq \mathbb{E}\Big[\|\nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y})\|_F \cdot \|Z'_{\ell,c} - Z_{\ell,c}\|_F\Big] \\ &\leq \sqrt{\mathbb{E}\Big[\|\nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y})\|_F^2\Big] \cdot \mathbb{E}\Big[\|Z'_{\ell,c} - Z_{\ell,c}\|_F^2\Big]} \end{aligned} \tag{5}$$

By optimizing the upper bound of the objective function, we avoid storing large amounts of the gradients of the feature maps or performing additional backpropagation. Substituting Equation (5) into the objective function to minimize Equation (4), we obtain

$$\min_{M^{t+1}} \sum_{\ell=1}^{L} \sum_{c=1}^{C_\ell} \mathbb{E}_{x\sim X_t^T} \|\nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y})\|_F^2 \cdot \|Z'_{\ell,c} - Z_{\ell,c}\|_F^2 \tag{6}$$

According to Figure 3, we observe an interesting phenomenon: The degree of unstable representation of the channels is positively correlated with the magnitude of the gradient values. Larger gradients cause the parameters to change rapidly, indicating that domain-sensitive parameters are sensitive to domain-specific knowledge and fit such knowledge through rapid changes, leading to unstable channel representations that are prone to variations and contain a large amount of domain-specific knowledge. This is consistent with the conclusion derived from the Equation (6): the larger the gradient value, the higher the degree of suppression of the parameters.

Therefore, we define a channel sensitivity importance weight as $I_{\ell,c}^t$:

$$I_{\ell,c}^t = \|\nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y})\|_F^2 \tag{7}$$

However, we cannot directly optimize using this equation. There are significant differences in the magnitude of channel gradients across layers, so it is necessary to balance the scale of importance across layers. Finally, we construct the channel sensitivity importance weight $I_{\ell,c}^t$ as follows:

$$I_{\ell,c}^t = \begin{cases} I_{\ell,c}^t = 0 & t = 0 \\ I_{\ell,c}^t = \frac{I_{\ell,c}^{t-1} + \|\nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y})\|_F^2}{\frac{1}{C_{\ell,c}} \sum_{c=1}^{C_\ell} \|\nabla_{Z_{\ell,c}} \mathcal{L}(G_\ell(Z_\ell), \hat{y})\|_F^2} & t > 0 \end{cases} \tag{8}$$

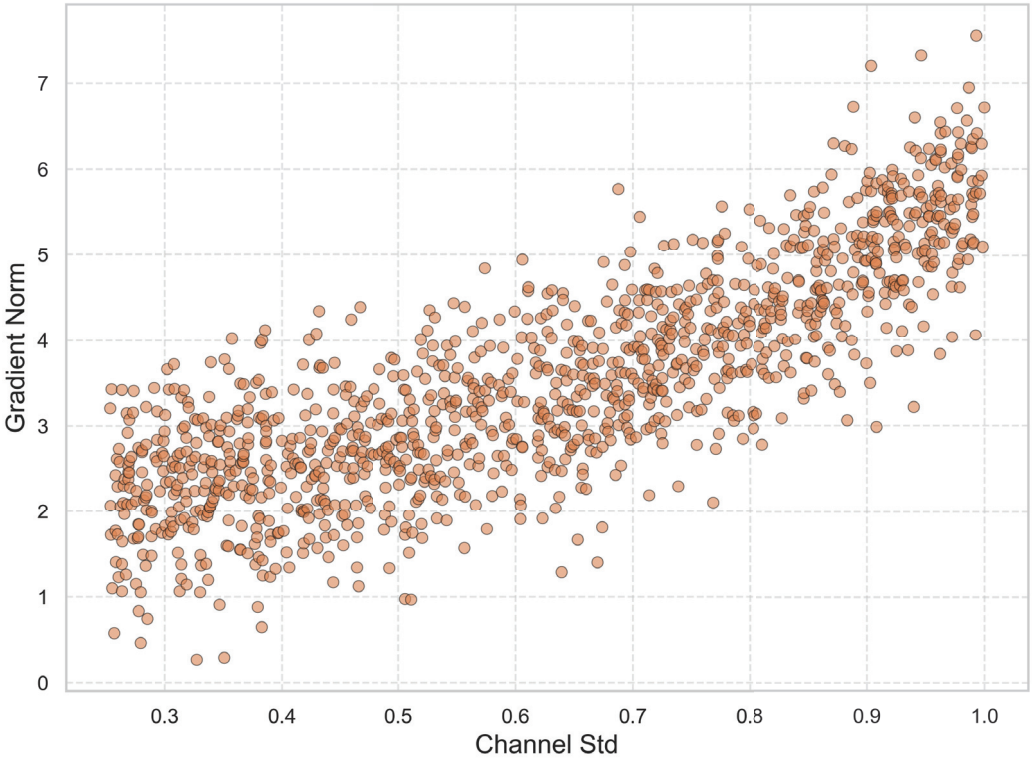


Figure 3. Gradient norm and channel-unstable representation relationship.

Therefore, the final parameter constraint objective function can be expressed as

$$\mathcal{L}_{DSP} = \min_{M^{t+1}} \sum_{\ell=1}^{L} \sum_{c=1}^{C_\ell} \mathbb{E}_{x \sim X_t^T} I_{\ell,c}^t \cdot \|Z'_{\ell,c} - Z_{\ell,c}\|_F^2 \tag{9}$$

3.2.3. Theoretical Analysis of Domain-Sensitive Parameter Suppression

In this section, we prove from the perspective of Lipschitz continuity that our method can effectively suppress domain-sensitive parameters, promote the learning of domain-invariant knowledge, and enhance the generalization ability of the model.

We first provide the definition of Lipschitz continuity. Given $\Omega \subset \mathbb{R}^n$, let $\theta_1 \in \Omega$ and $\theta_2 \in \Omega$. For a function $h : \Omega \to \mathbb{R}^m$, if there exists a constant K such that the following holds:

$$\|h(\theta_1) - h(\theta_2)\|_2 \leq K\|\theta_1 - \theta_2\|_2, \quad \forall \theta_1, \theta_2 \in \Omega \tag{10}$$

then, h is called Lipschitz continuous.

According to existing studies [33,47], if the loss function has a smaller Lipschitz constant K, it indicates that the loss function landscape is flatter, which consequently leads to better model generalization. On the contrary, if the Lipschitz constant K is very large, it indicates that the model parameters are highly sensitive to input variations, with even minor changes leading to drastic shifts in the model's output, which means that the model lacks generalization ability.

Now, consider the model parameters θ_{t-1} and θ_t at two consecutive moments during continual domain adaptation. The change in the loss function can be expressed as:

$$\|\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t-1})\|_2 = \|\mathcal{L}(\zeta)(\theta_t - \theta_{t-1})\|_2 \tag{11}$$

where $\zeta = c\theta_t + (1-c)\theta_{t-1}$, and $c \in [0,1]$. By applying the Cauchy-Schwarz inequality, we have

$$\|\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t-1})\|_2 \leq \|\nabla L(\zeta)\|_2 \|\theta_t - \theta_{t-1}\|_2 \tag{12}$$

Considering $\theta_t = \theta_{t-1} - \eta\nabla\theta_{t-1}$, we have $\theta_t \to \theta_{t-1}$, so $\nabla L(\zeta) \approx \nabla\mathcal{L}(\theta_{t-1})$, and the above can be rewritten as

$$\|\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t-1})\|_2 \leq \|\nabla\mathcal{L}(\theta_{t-1})\|_2 \|\theta_t - \theta_{t-1}\|_2 \tag{13}$$

From Equation (13), it can be seen that minimizing $\|\nabla\mathcal{L}(\theta_{t-1})\|_2$ is equivalent to minimizing the Lipschitz constant K. According to Equation (9), we have

$$\mathcal{L}_{DSP} \propto I^t_{\ell,c} = \|\nabla_{Z_{\ell,c}}\mathcal{L}(G_\ell(Z_\ell), \hat{y})\|^2_F \tag{14}$$

Thus, based on Equation (14), penalizing the gradient norm forces the model parameters to generate smaller gradient norms, which is equivalent to reducing the Lipschitz constant of the model. As a result, the model achieves better generalization performance.

3.3. Virtual Decision Boundary

3.3.1. Motivation of Virtual Decision Boundary

In Figure 4, we clarify the reasons behind the lack of robustness in the decision boundary and present our proposed solutions. In Figure 4a, we observe that during domain adaptation, the features extracted by the feature extractor $F(\cdot)$ tend to cluster near or slightly cross the decision boundary, which is indicated by the red boxes. In Figure 4b, we examine domain shifts, where we assume that two successive adapted domains have deviations ε_k and ε_{k+1} from the source domain. The deviation between them is $\Delta\varepsilon = \varepsilon_{k+1} - \varepsilon_k$. When $\Delta\varepsilon > 0$, the feature bias increases, leading to the possibility that features clustered near the decision boundary may cross over, resulting in errors. To address these challenges, we propose the virtual decision boundary. As shown in Figure 4c, this method introduces a virtual margin between the original decision boundary and a newly created virtual decision boundary. This virtual margin pushes features away from the original boundary, reducing the likelihood of clustering near it. Additionally, the virtual margin provides sufficient buffer space to help prevent features from crossing the decision boundary during domain shifts.

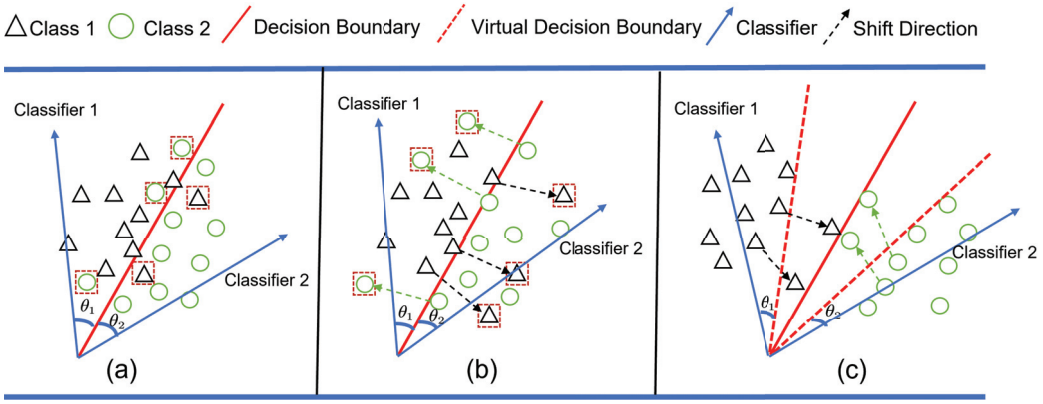


Figure 4. An illustrative example showcasing our motivation and solution: (**a**) During continuous domain adaptation, features are prone to cluster around or cross the decision boundary. (**b**) During domain shifts, features clustered at the decision boundary are prone to crossing it. (**c**) The virtual decision boundary forces the generated features away from the original decision boundary, creating a virtual margin that prevents features from crossing the decision boundary during domain shifts.

3.3.2. Implementation of Virtual Decision Boundary

Before addressing this issue, we start with a simple binary classification problem. Consider a binary classification problem where we have a sample from class 1 and use the feature extractor function $Z = F(x)$ to obtain the features of the sample x. We introduce a parameter m that is used to scale the inequality, forming a stricter virtual decision boundary. This can be mathematically expressed as $\|W_1\|_2\|Z\|_2\cos(m\theta_1) > \|W_2\|_2\|Z\|_2\cos(\theta_2)$ $(0 \leq \theta_1 \leq \frac{\pi}{m})$, where θ is the angle between the classifier vector and the feature vector, and $m \in [1, +\infty)$, as the following inequality holds:

$$\|W_1\|_2\|Z\|_2\cos(\theta_1) \geq \|W_1\|_2\|Z\|_2\cos(m\theta_1) > \|W_2\|_2\|Z\|_2\cos(\theta_2) \tag{15}$$

Therefore, $\|W_1\|_2\|Z\|_2\cos(\theta_1) > \|W_2\|_2\|Z\|_2\cos(\theta_2)$ has to hold. So the new classification criteria is a stronger requirement to correctly classify x, producing a more rigorous virtual decision boundary for class 1. Thus, the loss function for binary classification of class 1 can be written as

$$\mathcal{L} = \frac{e^{\|W_1\|_2\|Z\|_2\cos(m\theta_1)}}{e^{\|W_1\|_2\|Z\|_2\cos(m\theta_1)} + e^{\|W_2\|_2\|Z\|_2\cos(\theta_2)}} \tag{16}$$

We can also apply the same constraint to class 2 as we did for class 1. In CTTA, we can transform a binary classification problem into a multiclass classification problem.

$$\mathcal{L} = \frac{1}{|N|}\sum_{i=1}^{N}\frac{e^{\|W_c\|_2\|Z_i\|_2\cos(m\theta_c)}}{e^{\|W_c\|_2\|Z_i\|_2\cos(m\theta_c)} + \sum_{j\neq c}^{C} e^{\|W_j\|_2\|Z_i\|_2\cos(\theta_j)}} \tag{17}$$

In addition, in order to make Equation (17) hold for all $\theta \in \pi$, we construct $\psi(\theta)$ as follows:

$$\psi(\theta) = \begin{cases} \cos(m\theta) & 0 < \theta < \dfrac{\pi}{m} \\ \mathcal{D}(\theta) & \dfrac{\pi}{m} < \theta < \pi \end{cases} \tag{18}$$

where m is a non-negative number that is closely related to the classification margin. With larger m, the classification margin becomes larger, and the learning objective also becomes harder. Meanwhile, $\mathcal{D}(\theta)$ is required to be a monotonically decreasing function and $\mathcal{D}(\frac{\pi}{m})$ should equal $\cos(\frac{\pi}{m})$. We construct a specific $\psi(\theta)$ as follows:

$$\psi(\theta) = (-1)^k\cos(m\theta) - 2k, \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}] \tag{19}$$

where $k \in [0, m-1]$ and k is an integer. So, the final virtual decision boundary loss can be represented as follows:

$$\mathcal{L}_{VDB} = \frac{1}{|N|}\sum_{i=1}^{N}\frac{e^{\|W_c\|_2\|Z_i\|_2\psi(\theta_c)}}{e^{\|W_c\|_2\|Z_i\|_2\psi(\theta_c)} + \sum_{j\neq c}^{C} e^{\|W_j\|_2\|Z_i\|_2\cos(\theta_j)}} \tag{20}$$

3.3.3. Dynamic Virtual Margin of Virtual Decision Boundary

In the virtual decision boundary method, the width of the virtual margin m is not uniform across different domains and classes at any given time. This variability arises because different domains experience varying degrees of domain shift, and each class is affected differently, leading to varying densities of sample features near the decision boundary. To address this, we propose a dynamic margin that adjusts the width of the virtual margin according to the shift degree of each domain and class. A practical way to implement this is by using the classifier from the source model as a reference point. The deviation of each class from its corresponding class in the source domain can be

quantified using cosine distance. Additionally, the overall domain shift can be assessed by calculating the average cosine distance of all samples from their positions in the source domain. Specifically, we first measure the shift degree $\bar{D}$ of the current domain relative to the source domain:

$$\bar{D} = \frac{1}{2}\left(1 - \frac{1}{|N|}\sum_{i}^{N}\frac{W_c \cdot Z_i}{\|W_c\|_2\|Z_i\|_2}\right) \tag{21}$$

Then, we measure the shift degree D_c of each class relative to the corresponding class in the source domain:

$$D_c = \frac{1}{2}\left(1 - \frac{1}{|N_c|}\sum_{j}^{N_c}\frac{W_c \cdot Z_j}{\|W_c\|_2\|Z_j\|_2}\right), c \in C \tag{22}$$

Thus, at any given time t, the dynamic margin for each class in different domains can be expressed as

$$m_c^t = \begin{cases} m & t = 0 \\ \beta \cdot m_c^{t-1} + (1-\beta)\cdot\left(\lceil \frac{D_c}{\max(D)} \cdot e^{\bar{D}} + 1 \rceil\right), & t > 0 \end{cases} \tag{23}$$

where $D = \{D_1, D_2, \ldots, D_C\}$, and β is set to 0.999, used to robustly update, preventing its value from undergoing drastic changes. This dynamic adjustment allows the model to more effectively adapt to the specific shifts encountered across different domains and classes, enhancing its robustness and accuracy. The m is a hyperparameter used to initialize the virtual margin, and the specific setting is discussed in detail in the experimental section.

3.4. Loss Function

Based on Equations (1), (8) and (20), we can derive the following overall loss function:

$$\mathcal{L}_{all} = \mathcal{L}(M(X^T), \hat{Y}^T) + \mathcal{L}_{DSP} + \mathcal{L}_{VDB} \tag{24}$$

We constrain the model using this function to enhance its sustained capability.

4. Experiment and Results

4.1. Experimental Setup

4.1.1. Datasets and Task Setting

Building on the previous works [5,6,44], our method undergoes evaluation on three classification CTTA benchmarks, which encompass CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, and ImageNet-to-ImageNetC. In the segmentation CTTA, we conduct assessments on the Cityscapes-to-ACDC, using the Cityscapes [48] as the source domain and the ACDC [49] as the target domain.

4.1.2. Compared Methods and Implementation Details

We compare our method with the original model (Source) and multiple CTTA methods, including BN [50], TENT [12], CoTTA [5], RoTTA [44], SATA [51], RMT [6], PETAL [7], DSS [8], Reshaping [10]. For the classification task, all methods are implemented using the same backbone architecture and pretrained model as used in our approach. Specifically, we utilize the pretrained WideResNet-28 [52] for CIFAR10C, ResNeXt-29 [53] for CIFAR100C, and ResNet-50 [54] for ImageNetC, and use the largest corruption severity (level 5). For the segmentation CTTA task, we use the ACDC dataset as the target domain, which includes images captured under four distinct, unobserved visual conditions: Fog, Night, Rain, and Snow. To simulate continuous environmental changes akin to real-world scenarios, we

cyclically iterate through the same sequence of target domains (Fog → Night → Rain → Snow) multiple times.

4.2. Classification CTTA Tasks

4.2.1. CIFAR10-to-CIFAR10C Gradual and Continual

For the CIFAR10-to-CIFAR10C task, we evaluate our methods under two distinct settings. The first setting is a gradual task; the model sequentially adapts to fifteen target domains where the corruption severity level gradually changes between the lowest and highest extremes. The corruption type only changes when the severity reaches its lowest point. As shown in Table 1, our method achieves the lowest error rate of 8.3%, representing 2.1% improvement over the CoTTA method.

The second setting is a standard continual task where the model sequentially adapts to fifteen target domains, each with a corruption severity level of 5. The results, shown in Table 2, indicate that directly applying the source domain model yields an average error rate of 43.5%. The BN [50] method improves this performance by 23.1% compared with the source-only baseline. Among all compared methods, DSS achieves the lowest error rates of 12.2% on motion. SATA [51] achieves the lowest error rates of 10.2%, 14.1%, 13.2%, 10.3% on zoom, snow, frost, and contrast, respectively. Reshaping [10] achieves the lowest error rates of 17.1%, 12.7%, 15.9% on elastic, pixelate, and jpeg. In other scenarios, our proposed method either outperforms or is on par with the other approaches, ultimately achieving the lowest overall average error rate, reduced to 14.5%.

Table 1. Classification error rate (%) for the gradual CIFAR10-to-CIFAR10C task. The best results in each column are highlighted in **bold**.

Dataset	Source	BN [50]	TENT [12]	CoTTA [5]	Ours
CIFAR10C (Error %)	24.8	13.7	30.7	10.4	**8.3**

Table 2. Classification error rate (%) for the standard CIFAR10-to-CIFAR10C Continual Test-Time Adaptation task. The best results in each column are highlighted in **bold**.

	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixelate	Jpeg	Mean
Source	72.3	65.7	72.9	46.9	54.3	34.8	42.0	25.1	41.3	26.0	9.3	46.7	26.6	58.5	30.3	43.5
BN [50]	28.1	26.1	36.3	12.8	35.3	14.2	12.1	17.3	17.4	15.3	8.4	12.6	23.8	19.7	27.3	20.4
TENT [12]	24.8	20.6	28.6	14.4	31.1	16.5	14.1	19.1	18.6	18.6	12.2	20.3	25.7	20.8	24.9	20.7
CoTTA [5]	24.3	21.3	26.6	11.6	27.6	**12.2**	10.3	14.8	14.1	12.4	7.5	10.6	18.3	13.4	17.3	16.2
RoTTA [44]	30.3	25.4	34.6	18.3	34.0	14.7	11.0	16.4	14.6	14.0	8.0	12.4	20.3	16.8	19.4	19.3
RMT [6]	24.1	20.2	25.7	13.2	25.5	14.7	12.8	16.2	15.4	14.6	10.8	14.0	18.0	14.1	16.6	17.0
PETAL [7]	23.7	21.4	26.3	11.8	28.8	12.4	10.4	14.8	13.9	12.6	7.4	10.6	18.3	13.1	17.1	16.2
SATA [51]	23.9	20.1	28.0	11.6	27.4	12.6	**10.2**	**14.1**	**13.2**	12.2	7.4	**10.3**	19.1	13.3	18.5	16.1
DSS [8]	24.1	21.3	25.4	11.7	26.9	**12.2**	10.5	14.5	14.1	12.5	7.8	10.8	18.0	13.1	17.3	16.0
Reshaping [10]	23.6	19.9	26.0	11.8	25.3	13.2	10.9	14.3	13.5	12.7	9.0	11.9	**17.1**	**12.7**	**15.9**	15.8
Ours	**20.1**	**16.5**	**23.4**	**11.2**	**24.1**	12.6	10.3	14.3	13.6	**12.1**	**7.3**	10.9	18.2	12.9	17.0	**14.5**

4.2.2. CIFAR100-to-CIFAR100C

The results for the CIFAR100-to-CIFAR100C continual task, as shown in Table 3, further demonstrate the effectiveness of our method. Our approach achieves the lowest error rates across the Gaussian, shot, impulse, glass, motion, zoom, snow, frost, elastic, and jpeg, and it also obtains the lowest overall average error rate. Compared with the source-only baseline, our method improves performance by 17.9%, and it surpasses the Reshaping [10] method with a further 1.2% reduction in error rate.

Table 3. Classification error rate (%) for the standard CIFAR100-to-CIFAR100C Continual Test-Time Adaptation task. The best results in each column are highlighted in **bold**.

	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixelate	Jpeg	Mean
Source	73.0	68.0	39.4	29.3	54.1	30.8	28.8	39.5	45.8	50.3	29.5	55.1	37.2	74.7	41.2	46.4
BN [50]	42.1	40.7	42.7	27.6	41.9	29.7	27.9	34.9	35.0	41.5	26.5	30.3	35.7	32.9	41.2	35.4
TENT [12]	37.2	35.8	41.7	37.9	51.2	48.3	48.5	58.4	63.7	71.1	70.4	82.3	88.0	88.5	90.4	60.9
CoTTA [5]	40.1	37.7	39.7	26.9	38.0	27.9	26.4	32.8	31.8	40.3	24.7	26.9	32.5	28.3	33.5	32.5
RoTTA [44]	49.1	44.9	45.5	30.2	42.7	29.5	26.1	32.2	30.7	37.5	24.7	29.1	32.6	30.4	36.7	34.8
RMT [6]	40.2	36.2	36.0	27.9	33.9	28.4	26.4	28.7	28.8	31.1	25.5	27.1	28.0	26.6	29.0	30.2
PETAL [7]	38.3	36.4	38.6	25.9	36.8	27.3	25.4	32.0	30.8	38.7	24.4	26.4	31.5	26.9	32.5	31.5
SATA [51]	36.5	33.1	35.1	25.9	34.9	27.7	25.4	29.5	29.9	33.1	24.1	26.7	31.9	27.5	35.2	30.3
DSS [8]	39.7	36.0	37.2	26.3	35.6	27.5	25.1	31.4	30.0	37.8	24.2	26.0	30.0	26.3	31.1	30.9
Reshaping [10]	38.8	35.0	35.4	26.7	33.2	27.4	25.0	**27.4**	26.8	29.8	24.1	**25.1**	26.9	24.9	**28.0**	29.0
Ours	**33.9**	**32.6**	**31.8**	**25.4**	**30.2**	**26.7**	**24.8**	28.9	**24.3**	**29.7**	**23.4**	**25.1**	**26.5**	**23.3**	30.6	**27.8**

4.2.3. ImageNet-to-ImageNetC

Table 4 presents the performance comparison for various methods on the challenging ImageNet-to-ImageNetC continual task. Our method stands out by achieving the lowest average error rate among all the methods evaluated. Notably, it significantly outperforms the recently proposed Reshaping method across several difficult corruption types, including Gaussian (72.2% vs. 78.5%), shot (70.7% vs. 75.3%), impulse (68.3% vs. 73.0%), and glass (71.3% vs. 73.1%).

Table 4. Classification error rate (%) for the standard ImageNet-to-ImageNetC Continual Test-Time Adaptation task. The best results in each column are highlighted in **bold**.

	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixelate	Jpeg	Mean
Source	95.3	95.0	95.3	86.1	91.9	87.4	77.9	85.1	79.9	79.0	45.4	96.2	86.6	77.5	66.1	83.0
BN [50]	87.7	87.4	87.8	88.0	87.7	78.3	63.9	67.4	70.3	54.7	36.4	88.7	58.0	56.6	67.0	72.0
TENT [12]	81.6	74.6	72.7	77.6	73.8	65.5	55.3	61.6	63.0	51.7	38.2	72.1	50.8	47.4	53.3	62.6
CoTTA [5]	84.7	82.1	80.6	81.3	79.0	68.6	57.5	60.3	60.5	48.3	36.6	66.1	47.2	**41.2**	46.0	62.7
RoTTA [44]	88.3	82.8	82.1	91.3	83.7	72.9	59.4	66.2	64.3	53.3	**35.6**	74.5	54.3	48.2	52.6	67.3
RMT [6]	79.9	76.3	73.1	**75.7**	72.9	64.7	56.8	56.4	58.3	49.0	40.6	**58.2**	47.8	43.7	44.8	59.9
PETAL [7]	87.4	85.8	84.4	85.0	83.9	74.4	63.1	63.5	64.0	52.4	40.0	74.0	51.7	45.2	51.0	67.1
SATA [51]	74.1	72.9	71.6	**75.7**	74.1	64.2	55.5	55.6	62.9	46.6	36.1	69.9	50.6	44.3	48.5	60.1
DSS [8]	84.6	80.4	78.7	83.9	79.8	74.9	62.9	62.8	62.9	49.7	37.4	71.0	49.5	42.9	48.2	64.6
Reshaping [10]	78.5	75.3	73.0	**75.7**	73.1	64.5	56.0	55.8	**58.1**	47.6	38.5	58.5	**46.1**	42.0	43.4	59.0
Ours	**72.2**	**70.7**	**68.3**	75.9	**71.3**	**62.0**	**54.8**	**55.2**	61.3	**43.3**	40.5	61.2	48.6	42.1	**41.7**	**57.9**

4.3. Semantic Segmentation CTTA Task

Cityscapes-to-ACDC

We validate the effectiveness of our approach in the more challenging segmentation CTTA task by adapting the pretrained Segformer model from the Cityscapes dataset to the ACDC dataset, as shown in Table 5. Our method outperforms the previous entropy minimization method (TENT [12]), teacher–student method (CoTTA [5]), and sample reshape method (Reshaping [10] by 9.7%, 4.4%, and 0.4%, respectively). Notably, our method demonstrates better stability compared with others, with performance continuously improving throughout the adaptation process. This is attributed to the effective suppression of domain-sensitive parameters, which forces the model to learn more domain-invariant knowledge while mitigating the interference from domain-specific knowledge.

Table 5. Semantic segmentation results (mIoU in %) on the Cityscapes-to-ACDC CTTA task. The four test conditions are repeated ten times to evaluate the long-term adaptation performance. The best results in each column are highlighted in **bold**.

Time	$t \longrightarrow$																				
Round	**1**					**4**					**7**					**10**					**All**
Condition	**Fog**	**Night**	**Rain**	**Snow**	**Mean**	**Fog**	**Night**	**Rain**	**Snow**	**Mean**	**Fog**	**Night**	**Rain**	**Snow**	**Mean**	**Fog**	**Night**	**Rain**	**Snow**	**Mean**	**Mean ↑**
Source	69.1	40.3	59.7	57.8	56.7	69.1	40.3	59.7	57.8	56.7	69.1	40.3	59.7	57.8	56.7	69.1	40.3	59.7	57.8	56.7	56.7
BN	62.3	38.0	54.6	53.0	52.0	62.3	38.0	54.6	53.0	52.0	62.3	38.0	54.6	53.0	52.0	62.3	38.0	54.6	53.0	52.0	52.0
TENT [12]	69.0	40.2	60.1	57.3	56.7	66.5	36.3	58.7	54.0	53.9	64.2	32.8	55.3	50.9	50.8	61.8	29.8	51.9	47.8	47.8	52.3
CoTTA [5]	70.9	41.2	62.4	59.7	58.6	70.9	41.2	62.4	59.7	58.6	70.9	41.2	62.4	59.7	58.6	70.9	41.2	62.4	59.7	58.6	58.6
Reshaping [10]	71.2	42.3	65.0	62.0	60.1	**72.8**	43.6	66.7	63.3	61.6	72.5	42.5	66.8	63.3	61.3	72.5	42.9	66.7	63.0	61.3	61.3
ours	**71.8**	**43.1**	**65.2**	**62.3**	**60.6**	72.6	**44.6**	**66.7**	**63.5**	**61.9**	**72.8**	**43.5**	**67.8**	**63.4**	**61.9**	**73.1**	**43.9**	**67.3**	**64.0**	**62.1**	**61.7**

4.4. Ablation Study and Further Analysis

4.4.1. Ablation Study

We conducted an ablation study to assess the effectiveness of the key components of our approach across three benchmarks. For clarity, we refer to the virtual decision boundary as VDB and Domain Sensitivity Parameter Suppression as DSP. As shown in Table 6, incorporating VDB and DSP results in reduced error rates across all benchmarks. The combination of VDB and DSP leads to an even greater reduction in error rates, highlighting the synergistic effect of these components when used together.

Table 6. Ablation: Contribution of our proposed VDB and DSP. The best results in each column are highlighted in **bold**.

	VDB	**DSP**	**CIFAR10C**	**CIFAR100C**	**ImageNetC**
0			16.2%	32.5%	62.7%
1	✓		15.4%	30.1%	60.3%
2		✓	15.1%	28.3%	59.2%
3	✓	✓	**14.5%**	**27.8%**	**57.9%**

4.4.2. Integration with Existing Methods

Next, we integrate our method with existing methods, namely TENT [12], CoTTA [5], RMT [44], and DSS [8]. The experiments are conducted on the CIFAR10C and CIFAR100C datasets. Utilizing the official code of each method, we enhance the accuracy of all methods, as shown in Table 7. For example, our method reduces the error rate of CoTTA from 16.2% to 14.2% on CIFAR10C, from 32.5% to 26.9% on CIFAR100C, and from 62.6% to 57.8% on ImageNetC. These experiments and results demonstrate that our method can be seamlessly integrated with other CTTA methods to enhance performance.

Table 7. Integration with existing methods. Our method can be seamlessly integrated with other CTTA methods to boost performance.

Method	**CIFAR10C**	**CIFAR100C**	**ImageNetC**
TENT+ours	18.9% (+1.8%)	56.6% (+4.3%)	65.3% (+6.7%)
CoTTA+ours	14.2% (+2.0%)	26.9% (+5.6%)	57.8% (+4.8%)
RMT+ours	15.4% (+1.6%)	28.7% (+1.5%)	57.4% (+2.5%)
DSS+ours	14.1% (+1.7%)	26.1% (+4.8%)	58.5% (+6.1%)

4.4.3. Analysis of Domain Sensitivity Parameter Suppression

In this section, we focus on evaluating the effectiveness of Domain Sensitivity Parameter Suppression (DSP). The dataset used is CIFAR10C, and the pretrained network is

WideResNet-28. The methods compared include CoTTA [5], DSS [8], and RMT [6]. First, we measure the model's sensitivity to the domain by calculating the unstable representations of all channels in the network. The calculation formula is $Std_{channels} = \sum_{l=0}^{L} \sum_{i=0}^{C^l} \sqrt{\|\tilde{C}_i^l - C_i^l\|_2^2}$ where $\tilde{C}_i^l$ represent the channel features generated by the source domain data in the target domain network, and C_i^l represent the channel features generated by the target domain data in the target domain network. As shown in Figure 5, our method consistently achieves the lowest unstable activation values across all domains. This indicates that DSP effectively suppresses the learning of domain-specific knowledge by regulating the updates of domain-sensitive parameters. This, in turn, enhances the learning of domain-invariant knowledge and prevents the model from overfitting to current domain knowledge during continuous domain shifts, thereby mitigating interference with future domains that may be encountered. Furthermore, we examined the relationship between the number of channels in the first layer of the network and their corresponding unstable activation values. As shown in Figure 6, the unstable representations of DSP exhibit a smaller variance, with activation values concentrating around 0.35 across all channels. This demonstrates that the introduction of DSP can effectively control channel instability, enabling the model to maintain strong robustness and effectively counteract the effects of domain shifts.

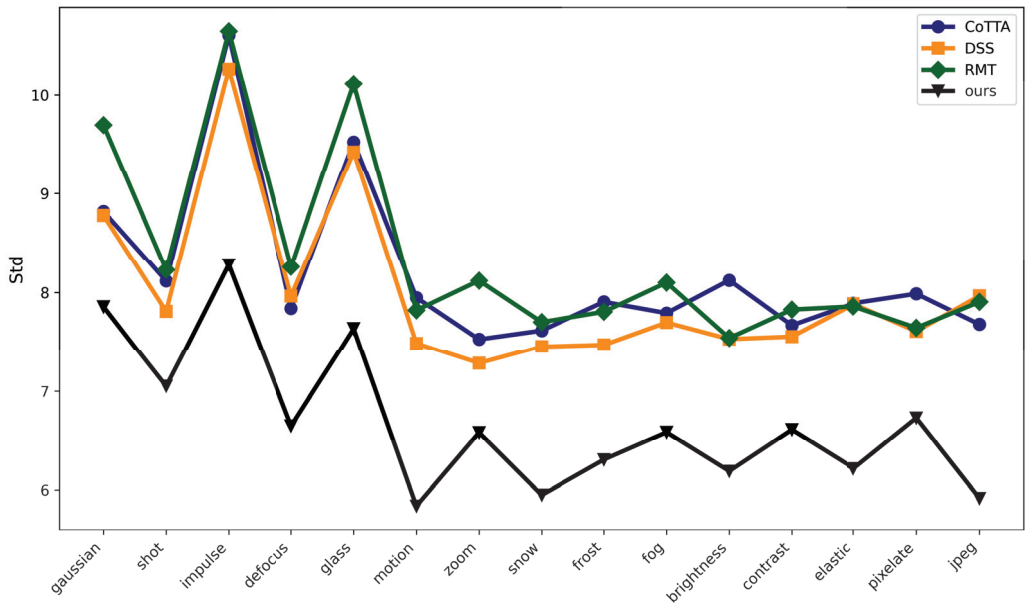


Figure 5. Channels std in all domains.

Additionally, inspired by adversarial training, we incorporate an auxiliary discriminator network to evaluate the effectiveness of DSP in promoting the learning of domain-invariant representations. Specifically, we add a source domain discriminator to the first layer of the WideResNet-28 network and train this discriminator on the CIFAR-10 dataset. The source domain discriminator takes the output of the first layer as input and generates a score indicating the likelihood that the input belongs to the source domain. Intuitively, if the domain-sensitive parameters are effectively suppressed, the model's output should remain highly robust and well aligned with the source domain, regardless of the domain shifts. As shown in Figure 7, compared with other methods on CIFAR10C, our approach consistently maintains high robustness to the source domain across all domains, indicating that the domain-sensitivity parameters have been effectively suppressed.

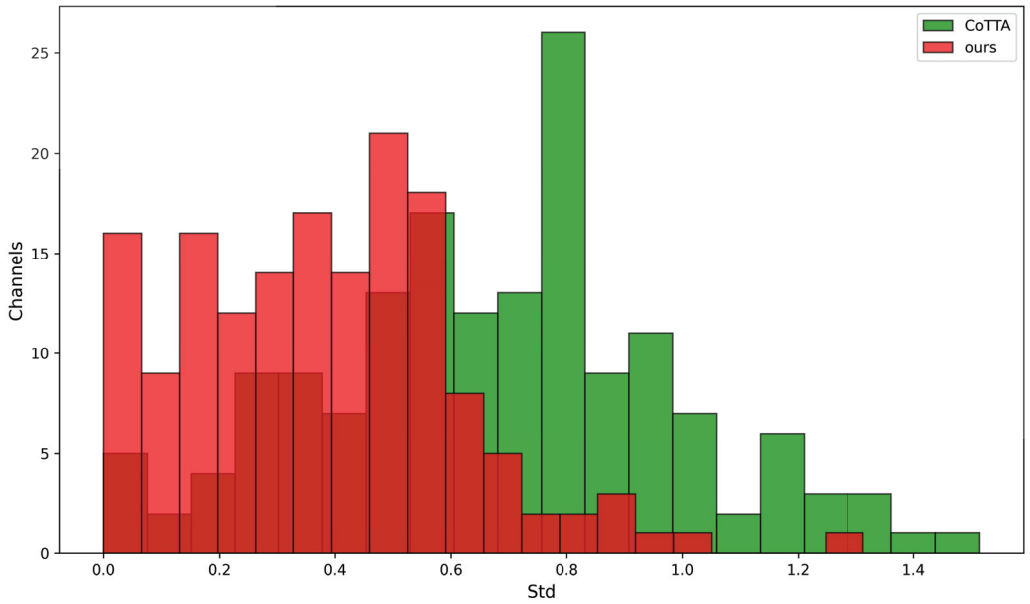


Figure 6. Channels std in first layer.

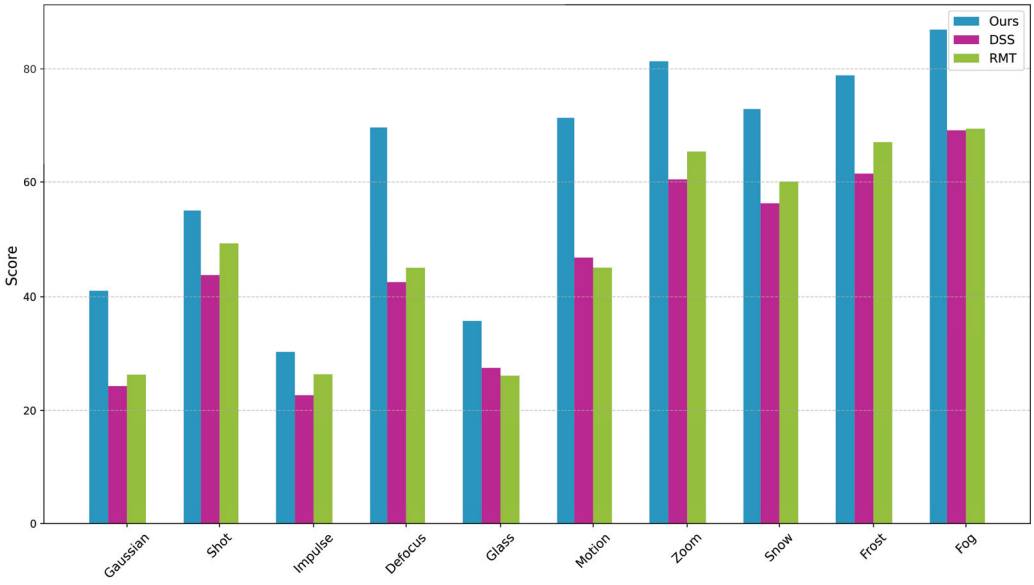


Figure 7. Domain robustness score.

4.5. Analysis of Dynamic Virtual Margin

In this section, we focus on discussing the parameter initialization of the virtual decision margin m and the superiority of dynamic virtual margins. First, we explore the effects of different initialization values for the parameters on three datasets (CIFAR10C, CIFAR100C, ImageNetC). As shown in Table 8, the initialization of the virtual decision margin parameters is highly robust. When m is set between 1 and 6, it performs well across all three datasets. However, when m exceeds 6, the performance begins to gradually decline. This is because a larger virtual margin increases the difficulty of model convergence during the initial phase of domain adaptation, leading to a drop in performance. We recommend initializing with a smaller virtual margin, allowing the margin to adaptively adjust to an appropriate value to avoid convergence difficulties caused by a large margin in the early stages of adaptation.

Overall, the dynamic virtual margin can adaptively adjust the margin width based on domain and class difficulty, and is not sensitive to initialization parameters, showing good robustness with excellent performance across a broad initialization range.

Next, we investigate the performance of fixed and dynamic virtual margins. As shown in Table 9, dynamic virtual margins outperform fixed virtual margins on all datasets. This is due to the dynamic margin's ability to adaptively adjust the margin size based on domain and class difficulty, achieving optimal performance.

Table 8. Performance results (error rate %) with different values of m on CIFAR10C, CIFAR100C, and ImageNetC datasets.

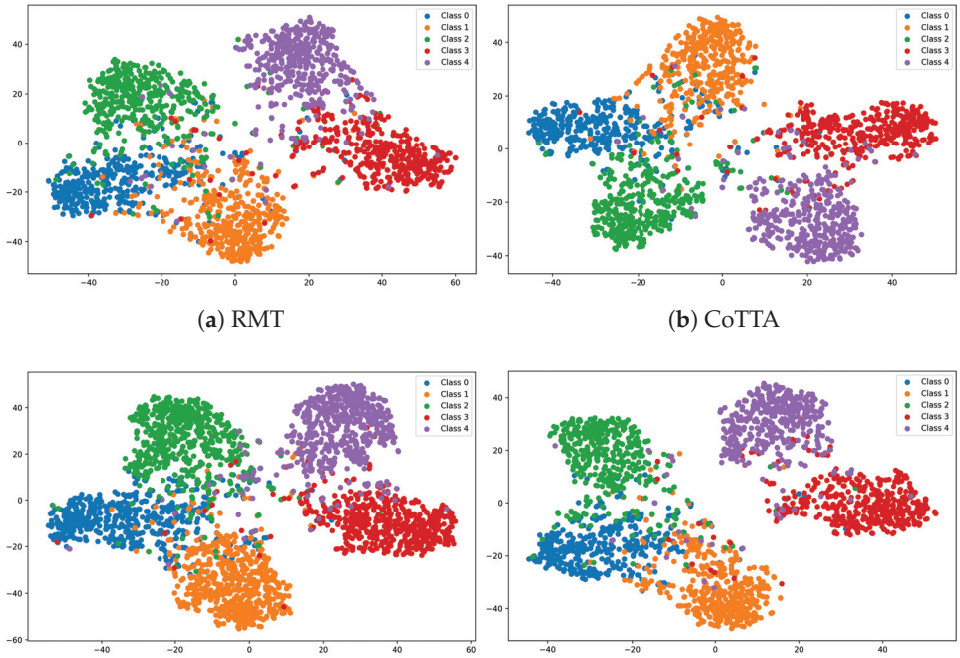
	m = 1	m = 2	m = 3	m = 4	m = 5	m = 6	m = 7	m = 8
CIFAR10C	14.8	14.5	14.7	14.9	15.3	15.5	16.1	16.8
CIFAR100C	27.9	27.8	28.2	28.3	28.4	28.7	29.5	30.1
ImageNetC	58.1	57.9	58.1	58.3	58.6	59.1	59.7	60.9

Table 9. Performance results (error rate %) with different values of fixed m and dynamic m on CIFAR10C, CIFAR100C, and ImageNetC datasets.

	m = 1	m = 2	m = 3	m = 4	m = 5	m = 6	m = 7	m = 8	m = Dynamic
CIFAR10C	15.3	15.2	14.9	15.5	15.9	16.3	16.7	17.3	14.5
CIFAR100C	28.2	28.1	28.5	28.9	29.3	29.7	30.2	31.1	27.8
ImageNetC	58.7	58.2	58.5	58.9	59.3	59.9	60.3	61.2	57.9

4.6. Analysis of Virtual Decision Boundary

We evaluate the effectiveness of our virtual decision boundary (VDB) method in mitigating feature crossing the decision boundary by analyzing inter-class and intra-class distances. Our VDB method is proposed to alleviate the issue of feature clustering at or slightly crossing the decision boundary. First, we utilized T-SNE to visualize the sample features generated by three different methods (DSS [8], CoTTA [5], RMT [6]) and our method in the Gaussian domain of the CIFAR10C dataset. As shown in Figure 8, the results demonstrate that the features generated by our method exhibit better clustering performance, with smaller intra-class distances and larger inter-class distances. Second, if class feature shifts are well controlled, the inter-class distance should increase, and the intra-class distance should decrease. This will effectively prevent features from crossing the decision boundary. The intra-class distance is expressed as $d_{intra} = \sum_{i}^{C} ||z_i - \overline{z_i}||_2$, and the inter-class distance is expressed as $d_{inter} = \sum_{i}^{C} \sum_{j \neq i}^{C} ||z_i - \overline{z_j}||_2$, where $\overline{z}$ represents the mean feature of the class. As shown in Figures 9 and 10, we compare our approach with three other methods include DSS [8], CoTTA [5], and RMT [44]. The results demonstrate that our method excels in managing class shifts during CTTA, consistently achieving more desirable intra-class and inter-class distances across all domains. In particular, to evaluate the effectiveness of our virtual decision boundary in mitigating feature crossing the decision boundary during the early stages of domain adaptation, we counted the number of features crossing the decision boundary for each method at the early stage of domain adaptation. As shown in Figure 11, our method consistently outperforms others across all domains.



(**a**) RMT (**b**) CoTTA

(**c**) DSS (**d**) Ours

Figure 8. Visualization on the Gaussian domain of the CIFAR10C dataset.

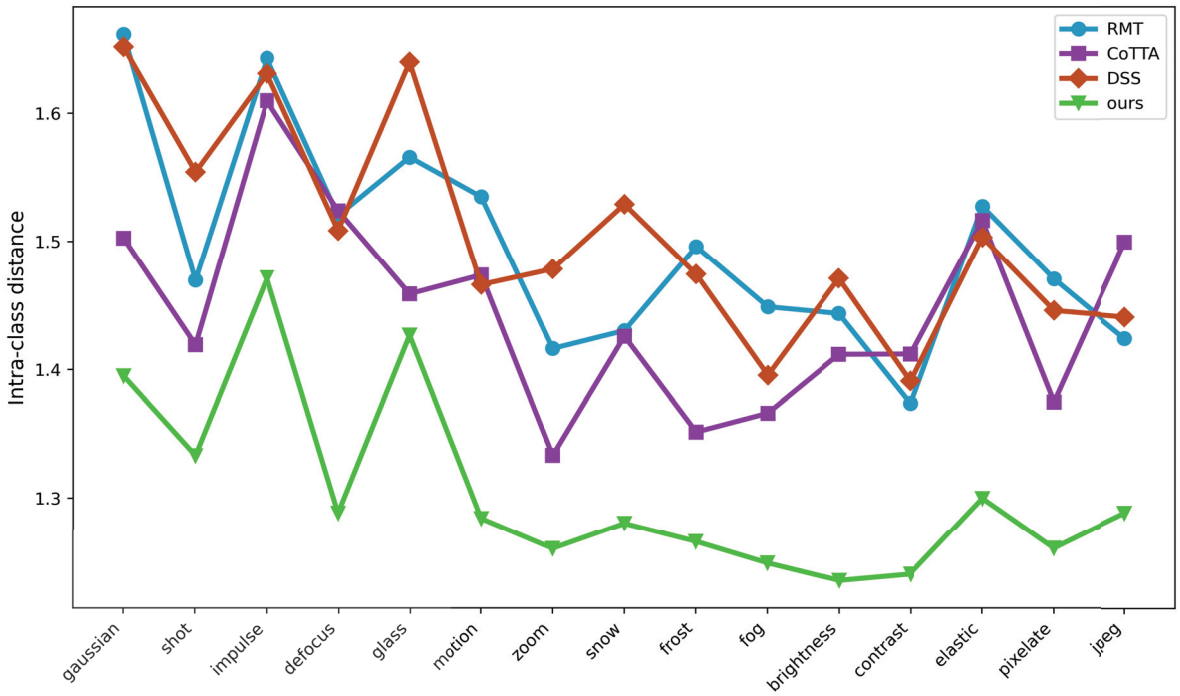


Figure 9. Intra-class distance in all domains.

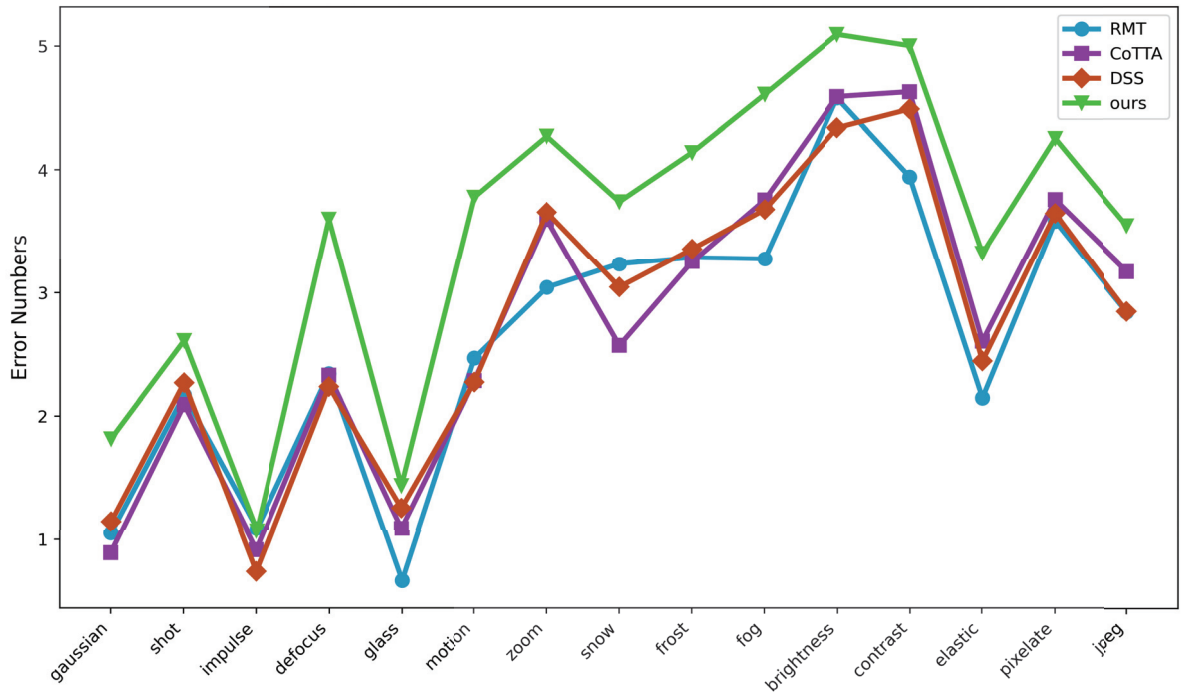


Figure 10. Inter-class distance in all domains.

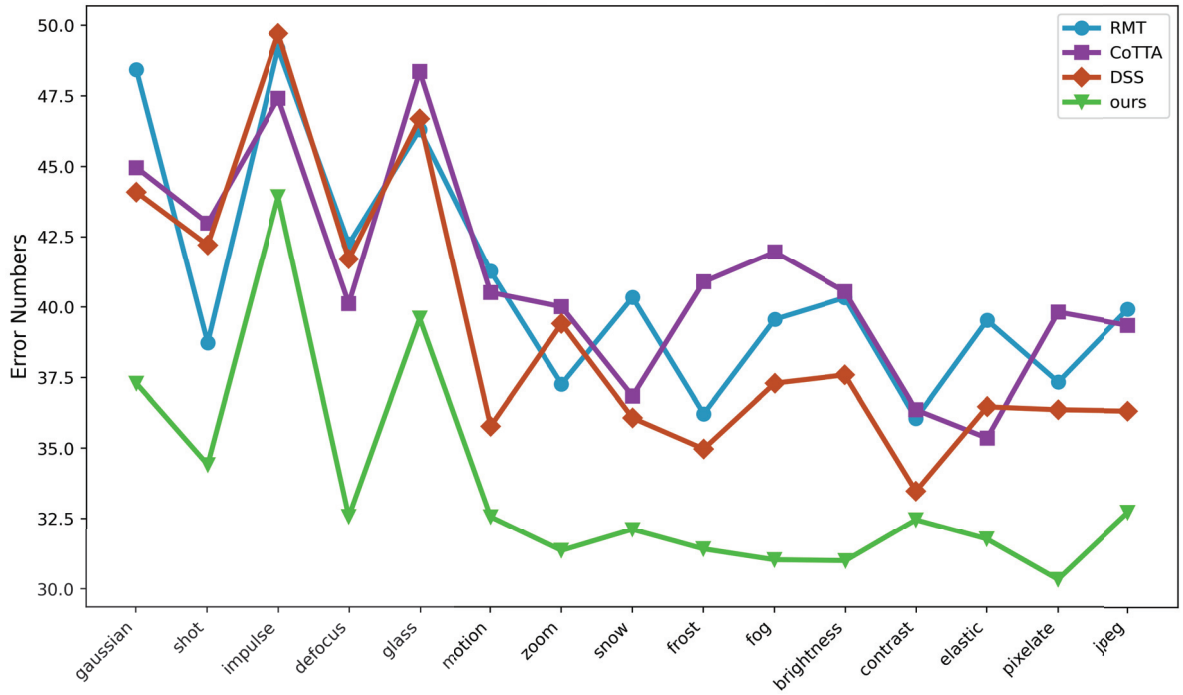


Figure 11. Error numbers at the domain begin.

4.7. Time and Parameter Complexity Evaluation

In this section, we focus on analyzing the time and space complexity of our method compared with other methods. The datasets used include CIFAR10C, CIFAR100C, and ImageNetC, and the methods for comparison include TENT [12], CoTTA [5], DSS [8], and RMT [6]. As shown in Table 10, the time required by our method is only 3 min, 4 min, and 13 min more than the most time-consuming DSS [8] method on the CIFAR10C, CIFAR100C, and ImageNetC datasets, respectively. The reason for the additional time

consumption is that our method requires the computation of the channel feature map from the previous model (without computing gradients) and the calculation of the sensitivity loss function. As shown in Figure 12, compared with CoTTA [5], DSS [8], and RMT [6], our method requires more GPU memory at runtime, approximately 0.8GB, due to the additional storage of feature maps computed by the previous model. This additional time and space complexity overhead are small and can be largely ignored, which leads to significant performance improvements.

Table 10. Time required for different methods on CIFAR10C, CIFAR100C, and ImageNetC datasets.

	TENT	CoTTA	DSS	RMT	Ours
CIFAR10C	7 min	15 min	16 min	15 min	19 min
CIFAR100C	9 min	17 min	19 min	18 min	23 min
ImageNetC	33 min	71 min	73 min	72 min	86 min

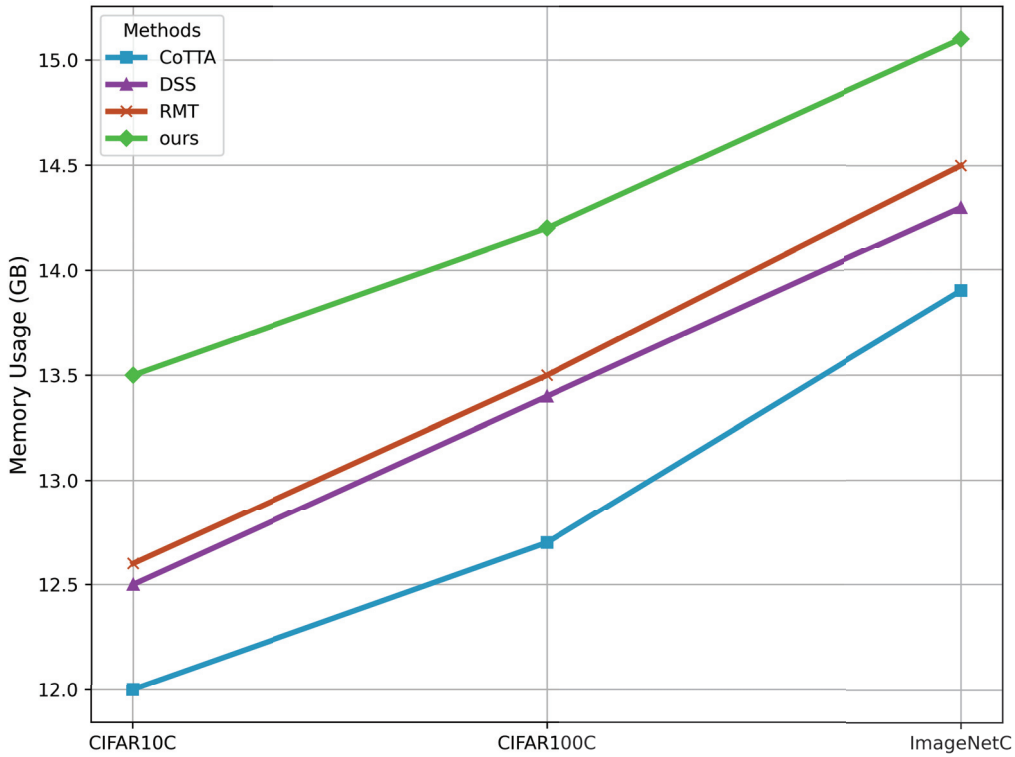


Figure 12. Memory usage for different methods on various datasets.

5. Conclusions

Distinguishing itself from previous research that tends to concentrate on isolated domain adaptation and seeks local optima. Instead, this paper focuses on the often overlooked sustained capability in Continual Test-Time Adaptation. To address this issue, we propose a Dual Constraint method that combines parameter constraints based on channel representations and virtual decision boundary constraints based on features. The parameter constraint penalizes the model's reliance on domain-specific knowledge, forcing it to learn domain-invariant knowledge while alleviating channel instability and mutual interference caused by overfitting to domain-sensitive parameters. This approach also theoretically enhances the model's generalization ability. Additionally, we provide theoretical evidence that our method can effectively enhance the model's generalization ability and promote the learning of domain-invariant knowledge. Meanwhile, the virtual decision boundary constraint pushes features away from the original decision boundary, forming a virtual margin

that buffers against domain shifts, effectively reducing the mutual interference between domain-specific knowledge. Our extensive experiments on multiple CTTA benchmark datasets have demonstrated the effectiveness of our proposed methods.

Author Contributions: Conceptualization, P.L. and Y.W.; Methodology, Y.S. and P.L.; Software, P.L.; Investigation, Y.W.; Writing—review & editing, Y.S. and P.L.; Visualization, Y.W.; Supervision, Y.S.; Funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under the Grant No. 62002330.

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

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

References

1. Hu, C.; Hudson, S.; Ethier, M.; Al-Sharman, M.; Rayside, D.; Melek, W. Sim-to-real domain adaptation for lane detection and classification in autonomous driving. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 4–9 June 2022; IEEE: New York, NY, USA, 2022; pp. 457–463.
2. Shi, Z.; Su, T.; Liu, P.; Wu, Y.; Zhang, L.; Wang, M. Learning Frequency-Aware Dynamic Transformers for All-In-One Image Restoration. *arXiv* **2024**, arXiv:2407.01636. [CrossRef]
3. Chen, W.; Miao, L.; Gui, J.; Wang, Y.; Li, Y. FLsM: Fuzzy Localization of Image Scenes Based on Large Models. *Electronics* **2024**, *13*, 2106. [CrossRef]
4. Xu, E.; Zhu, J.; Zhang, L.; Wang, Y.; Lin, W. Research on Aspect-Level Sentiment Analysis Based on Adversarial Training and Dependency Parsing. *Electronics* **2024**, *13*, 1993. [CrossRef]
5. Wang, Q.; Fink, O.; Van Gool, L.; Dai, D. Continual test-time domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7201–7211.
6. Döbler, M.; Marsden, R.A.; Yang, B. Robust mean teacher for continual and gradual test-time adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7704–7714.
7. Brahma, D.; Rai, P. A probabilistic framework for lifelong test-time adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 3582–3591.
8. Wang, Y.; Hong, J.; Cheraghian, A.; Rahman, S.; Ahmedt-Aristizabal, D.; Petersson, L.; Harandi, M. Continual test-time domain adaptation via dynamic sample selection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2024; pp. 1701–1710.
9. Liu, J.; Yang, S.; Jia, P.; Zhang, R.; Lu, M.; Guo, Y.; Xue, W.; Zhang, S. Vida: Homeostatic visual domain adapter for continual test time adaptation. *arXiv* **2023**, arXiv:2306.04344.
10. Zhu, Z.; Hong, X.; Ma, Z.; Zhuang, W.; Ma, Y.; Dai, Y.; Wang, Y. Reshaping the Online Data Buffering and Organizing Mechanism for Continual Test-Time Adaptation. In Proceedings of the European Conference on Computer Vision, Dublin, Ireland, 22–23 October 2025; Springer: Berlin/Heidelberg, Germany, 2025; pp. 415–433.
11. Liu, J.; Xu, R.; Yang, S.; Zhang, R.; Zhang, Q.; Chen, Z.; Guo, Y.; Zhang, S. Continual-MAE: Adaptive Distribution Masked Autoencoders for Continual Test-Time Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 28653–28663.
12. Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; Darrell, T. Tent: Fully test-time adaptation by entropy minimization. *arXiv* **2020**, arXiv:2006.10726.
13. DeVore, R.A.; Temlyakov, V.N. Some remarks on greedy algorithms. *Adv. Comput. Math.* **1996**, *5*, 173–187. [CrossRef]
14. Knowles, J.D.; Watson, R.A.; Corne, D.W. Reducing local optima in single-objective problems by multi-objectivization. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Zurich, Switzerland, 7–9 March 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 269–283.
15. Wang, D.Z.; Lo, H.K. Global optimum of the linearized network design problem with equilibrium flows. *Transp. Res. Part B Methodol.* **2010**, *44*, 482–492. [CrossRef]
16. Zhou, W.; Zhou, Z. Unsupervised Domain Adaption Harnessing Vision-Language Pre-Training. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 8201–8214. [CrossRef]
17. Wilson, G.; Cook, D.J. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol. (Tist)* **2020**, *11*, 1–46. [CrossRef]

18. Long, M.; Cao, Y.; Wang, J.; Jordan, M. Learning transferable features with deep adaptation networks. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; PMLR: San Diego, CA, USA, 2015; pp. 97–105.
19. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Unsupervised domain adaptation with residual transfer networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
20. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep transfer learning with joint adaptation networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: San Diego, CA, USA, 2017; pp. 2208–2217.
21. Fan, X.; Wang, Q.; Ke, J.; Yang, F.; Gong, B.; Zhou, M. Adversarially adaptive normalization for single domain generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8208–8217.
22. Yang, F.E.; Cheng, Y.C.; Shiau, Z.Y.; Wang, Y.C.F. Adversarial teacher-student representation learning for domain generalization. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 19448–19460.
23. Zhu, W.; Lu, L.; Xiao, J.; Han, M.; Luo, J.; Harrison, A.P. Localized adversarial domain generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7108–7118.
24. Liang, J.; Hu, D.; Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; PMLR: San Diego, CA, USA, 2020; pp. 6028–6039.
25. Lowd, D.; Meek, C. Adversarial learning. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 21–24 August 2005; pp. 641–647.
26. Gopnik, A.; Glymour, C.; Sobel, D.M.; Schulz, L.E.; Kushnir, T.; Danks, D. A theory of causal learning in children: Causal maps and Bayes nets. *Psychol. Rev.* **2004**, *111*, 3. [CrossRef] [PubMed]
27. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169. [CrossRef] [PubMed]
28. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 702–703.
29. Zhou, K.; Yang, Y.; Hospedales, T.; Xiang, T. Deep domain-adversarial image generation for domain generalisation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13025–13032.
30. Gan, Y.; Bai, Y.; Lou, Y.; Ma, X.; Zhang, R.; Shi, N.; Luo, L. Decorate the newcomers: Visual domain prompt for continual test time adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 7595–7603.
31. Wang, H.; Ge, S.; Lipton, Z.; Xing, E.P. Learning robust global representations by penalizing local predictive power. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
32. Guo, J.; Qi, L.; Shi, Y. Domaindrop: Suppressing domain-sensitive channels for domain generalization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 19114–19124.
33. Zhao, Y.; Zhang, H.; Hu, X. Penalizing gradient norm for efficiently improving generalization in deep learning. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; PMLR: San Diego, CA, USA, 2022; pp. 26982–26992.
34. Cong, W.; Cong, Y.; Sun, G.; Liu, Y.; Dong, J. Self-Paced Weight Consolidation for Continual Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 2209–2222. [CrossRef]
35. Zhang, W.; Huang, Y.; Zhang, W.; Zhang, T.; Lao, Q.; Yu, Y.; Zheng, W.S.; Wang, R. Continual Learning of Image Classes With Language Guidance From a Vision-Language Model. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 13152–13163. [CrossRef]
36. Yu, D.; Zhang, M.; Li, M.; Zha, F.; Zhang, J.; Sun, L.; Huang, K. Contrastive Correlation Preserving Replay for Online Continual Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 124–139. [CrossRef]
37. Li, H.; Liao, L.; Chen, C.; Fan, X.; Zuo, W.; Lin, W. Continual Learning of No-Reference Image Quality Assessment With Channel Modulation Kernel. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 13029–13043. [CrossRef]
38. Li, K.; Chen, H.; Wan, J.; Yu, S. ESDB: Expand the Shrinking Decision Boundary via One-to-Many Information Matching for Continual Learning With Small Memory. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 7328–7343. [CrossRef]
39. Shi, Z.; Liu, P.; Su, T.; Wu, Y.; Liu, K.; Song, Y.; Wang, M. Densely Distilling Cumulative Knowledge for Continual Learning. *arXiv* **2024**, arXiv:2405.09820. [CrossRef]
40. Wu, Y.; Chi, Z.; Wang, Y.; Plataniotis, K.N.; Feng, S. Test-time domain adaptation by learning domain-aware batch normalization. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March; Volume 38, pp. 15961–15969.
41. Zhang, J.; Qi, L.; Shi, Y.; Gao, Y. Domainadaptor: A novel approach to test-time adaptation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 18971–18981.

42. Chen, K.; Gong, T.; Zhang, L. Camera-Aware Recurrent Learning and Earth Mover's Test-Time Adaption for Generalizable Person Re-Identification. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 357–370. [CrossRef]
43. Wang, Y.; Chen, H.; Heng, Q.; Hou, W.; Fan, Y.; Wu, Z.; Wang, J.; Savvides, M.; Shinozaki, T.; Raj, B.; et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv* **2022**, arXiv:2205.07246.
44. Yuan, L.; Xie, B.; Li, S. Robust test-time adaptation in dynamic scenarios. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 15922–15932.
45. Shi, B.; Zhang, D.; Dai, Q.; Zhu, Z.; Mu, Y.; Wang, J. Informative dropout for robust representation learning: A shape-bias perspective. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; PMLR: San Diego, CA, USA, 2020; pp. 8828–8839.
46. Klinker, F. Exponential moving average versus moving exponential average. *Math. Semesterber.* **2011**, *58*, 97–107.
47. Dinh, L.; Pascanu, R.; Bengio, S.; Bengio, Y. Sharp minima can generalize for deep nets. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: San Diego, CA, USA, 2017; pp. 1019–1028.
48. Yang, S.; Wu, J.; Liu, J.; Li, X.; Zhang, Q.; Pan, M.; Zhang, S. Exploring sparse visual prompt for cross-domain semantic segmentation. *arXiv* **2023**, arXiv:2303.09792.
49. Sakaridis, C.; Dai, D.; Van Gool, L. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10765–10775.
50. Li, Y.; Wang, N.; Shi, J.; Liu, J.; Hou, X. Revisiting batch normalization for practical domain adaptation. *arXiv* **2016**, arXiv:1603.04779. [CrossRef]
51. Chakrabarty, G.; Sreenivas, M.; Biswas, S. Sata: Source anchoring and target alignment network for continual test time adaptation. *arXiv* **2023**, arXiv:2304.10113. [CrossRef]
52. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146. [CrossRef]
53. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
54. 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, 26 June–1 July 2016; pp. 770–778.

electronics

MDPI

Article

Joint Event Detection with Dynamic Adaptation and Semantic Relevance

Xi Zeng [1,2,*], Guangchun Luo [1] and Ke Qin [1]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; gcluo@uestc.edu.cn (G.L.); qinke@uestc.edu.cn (K.Q.)
[2] The 30th Research Institute of China Electronics Technology Group Corporation, Chengdu 610000, China
* Correspondence: 202112090811@std.uestc.edu.cn

Abstract: Event detection is a crucial task in natural language processing, and it plays a significant role in numerous applications, such as information retrieval, question answering, and situational awareness. Real-world tasks typically require robust models that can dynamically adapt to changing data distributions and seamlessly accommodate emerging event types while maintaining high accuracy and efficiency. However, existing methods often face catastrophic forgetting, a significant challenge where models lose previously acquired knowledge when learning new information. This phenomenon hinders models from balancing performance with adaptability, limiting their ability to generalize across dynamic data landscapes. This paper proposes a novel event detection framework, DASR, which aims to enhance the flexibility and diversity of event detection through joint learning and guidance that dynamically adapts to new events and extracts semantic relevance. Firstly, we utilize pre-trained language models (PLMs) trained on general corpora to obtain existing event and type information as global knowledge. Secondly, during prompt fine-tuning for specific tasks, we incorporate an incremental learning module to design incremental prompt templates for newly introduced event types and read out their representations within the PLM. Subsequently, we perform entity recognition and event trigger word detection to extract semantic relevance. In this case, a graph attention mechanism is introduced to enhance the long-distance dependencies within the text (modeled as message passing in the type graph). Additionally, feature fusion integrates entity and event trigger word information into a unified representation. Finally, we validate the effectiveness of the proposed framework through extensive experiments. The experimental results demonstrate that the proposed framework effectively mitigates catastrophic forgetting and significantly improves the accuracy and adaptability of event detection when dealing with evolving data distributions and newly introduced event types.

Keywords: event detection; incremental learning; pre-trained language model; graph neural networks

1. Introduction

In recent years, event detection and extraction [1] have garnered significant attention within the natural language processing (NLP) community [2,3]. These tasks are pivotal for understanding textual information, as they enable the identification and analysis of events from textual data. This, in turn, facilitates numerous downstream applications, such as information retrieval [4], question answering [5], and real-time decision-making systems [6]. Existing works have made considerable progress in this domain, focusing

on enhancing model accuracy through advanced techniques such as deep learning [7], attention mechanisms [8], and multi-task learning [9]. Despite these advancements, the dynamic nature of real-world applications presents a common challenge, i.e., the continual emergence and evolution of new events [10–13]. The constantly evolving data distribution and the emergence of novel event types pose significant challenges in maintaining detection accuracy and adaptability [14,15]. These challenges often lead to catastrophic forgetting [16,17], where models fail to retain previously learned knowledge effectively when acquiring new information. Continuous innovation is essential for building resilient event detection systems in dynamic environments.

Incremental learning (IL) [18–22] has been widely employed to address issues related to evolving and expanding datasets. This approach allows models to update their knowledge base without retraining from scratch, preserving old knowledge while learning new information. This characteristic is particularly beneficial for tasks requiring continual adaptation to new data [23]. However, employing IL for event detection [24–26] raises unique and multifaceted challenges. For instance, a key challenge lies in retaining the capability to accurately recognize previously known events with high fidelity, while simultaneously identifying and integrating newly emerging events. Furthermore, this process must skillfully manage the intricate temporal and causal relationships between events, which inevitably increases the complexity of detection methods. Pre-trained language models (PLMs) [27–29] trained on diverse language tasks provide an effective starting point. This extensive training endows them with a profound understanding of language, allowing for fine-tuning on specific tasks such as event detection. The inherent versatility and robust linguistic comprehension of PLMs make them particularly well-suited for IL scenarios, emphasizing the critical need to adapt dynamically without catastrophic forgetting. As such, leveraging PLMs in the realm of IL for event detection holds significant promise, providing a foundation upon which more sophisticated and adaptive models can be developed to tackle the ever-evolving landscape of textual information.

In addition to providing the necessary context information, PLMs need to explore deeper event information to achieve target detection as the diversity of events increases. Existing event detection research focuses on treating multiple event instances as independent data instances and identifying them one by one [11,30–33]. However, these approaches overlook the interactive correlations among event instances within a sentence. This concept, inspired by message aggregation in graph neural networks (GNNs) [34–39], can empirically offer further semantic representation guidance for event detection. GNNs are models designed to process data structured as graphs, enabling the capture of relationships and dependencies between nodes. As shown in Figure 1, compared to traditional event detection models, our model enhances adaptability to new events and extracts useful information from complex correlations through prompt-based incremental fine-tuning and event correlation modeling, rather than being limited to inherent knowledge. In summary, **integrating dynamic event adaptation and diverse semantic relevance aims to address dual challenges: maintaining accuracy within evolving data distributions and capturing the relationships between new event types and existing events.**

To address these challenges, we propose a novel joint event detection method with dynamic adaptation and semantic relevance, named **DASR**, which consists of several key components that are meticulously designed to tackle the unique problems inherent in event detection. First, during the pretraining phase, we utilize PLMs to establish a preliminary and comprehensive understanding of language from general corpora, laying a solid foundation for subsequent tasks. During the fine-tuning phase for specific tasks, we introduce

an innovative incremental learning module and design prompt templates to effectively model events. Additionally, to simultaneously handle entity recognition and event trigger word detection, we model the graph structure of event types based on word importance and event-type relevance. We employ an attention mechanism—a model component that assigns varying importance to different parts of the input text—to capture dependencies, facilitating a deeper understanding of the context. We validate this framework through extensive experiments, demonstrating the performance of DASR in mitigating catastrophic forgetting and enhancing adaptability to new event types. DASR's key innovation lies in its combination of prompt-based incremental learning and semantic relevance modeling using graph neural networks. Unlike traditional event detection systems, which often struggle to adapt to evolving event types, DASR's approach allows for continuous adaptation to new events, significantly reducing the risk of catastrophic forgetting. Furthermore, the use of GNNs enables DASR to capture complex event relationships that are typically overlooked by conventional methods, leading to more accurate and robust event detection in dynamic environments.

- We propose a novel framework that seamlessly integrates prompt-based incremental learning and semantic relevance in event detection, thereby enhancing the accuracy and adaptability of event detection.
- Through prompt engineering, type relevance modeling, and disentangled prediction of type interactions, we capture dynamic event relationships, quantify event relevance to provide interpretability for predictions, and mitigate the challenges of entity recognition with long-distance dependencies.
- Extensive experiments show that DASR improves event detection performance, mitigates catastrophic forgetting, and adapts to evolving data distributions.

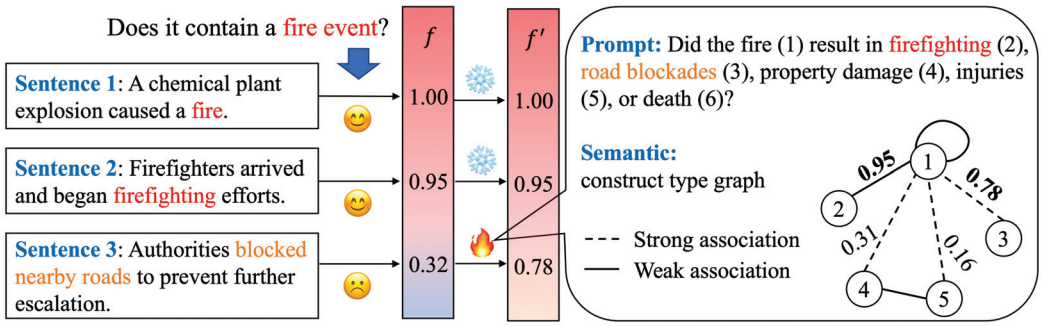


Figure 1. A toy example of event detection. Traditional model f struggles to capture new event correlations and makes erroneous judgments when dealing with temporal context. The improved model f' enhances detection accuracy through incremental prompting and semantic relevance.

The organization of this paper is organized as follows: We describe the overall proposed method in detail and explain the key components and techniques employed in our approach in Section 2. We present and analyze the experimental results, demonstrating the effectiveness of our method through various evaluations and comparisons in Section 3. Finally, we summarize the key findings and contributions of our work and discuss potential future research directions in Section 4.

2. Materials and Methods

2.1. Preliminaries

This section outlines the key concepts and methodologies underpinning DASA, which integrates state-of-the-art techniques in event detection and incremental learning. It incorporates LLMs for their rich contextual representations, memory-augmented networks for knowledge retention, and prompt learning for rapid adaptation to new event types.

Event detection. Event detection is a critical task within NLP that is focused on identifying events and their attributes in textual data. Traditional approaches have relied on static models that struggle to adapt to evolving data distributions and new event types. We aim to capture better syntactic and semantic features for more accurate event detection. Given a sequence of textual input, the task is to identify and classify event triggers and their associated arguments. Formally, let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ represent the event types, and $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ denote several candidate triggers. The objective is to predict a set of events $E = \{e_1, e_2, \ldots, e_n\}$, where each event e_i consists of a true event trigger $\mathbf{x}_i$ and event type t.

Incremental learning. The incremental learning component seeks to adapt the event detection model to new data without requiring retraining from scratch, all while preserving previously learned knowledge. We aim to develop a robust framework for event detection that leverages incremental learning to maintain performance in dynamic environments. For each new dataset, $D' = (X', Y')$, the model should update its parameters to minimize the loss on D', denoted as $\mathcal{L}(X', Y'; \theta)$, while minimizing the forgetting factor on previous datasets D. In addition, the IL aims to update the model $f_\theta : ([\mathbf{x}; \mathbf{x}'] \rightarrow [\mathbf{y}; \mathbf{y}'] \in Y)$ from the sequence of events.

Prompt engineering. This is also known as contextual prompting, a method that guides LLMs to locate knowledge relevant to specific targets without updating the model's weights or parameters. Formally, the prompt function $x' = g(x)$ is applied to the input x to obtain the corresponding prompt x', which includes x, the intermediate answer $\mathbf{z}$, and discrete or continuous task-specific tokens as task descriptors. Given a prompt, a PLM model can generate the answer $\mathbf{z}$. Prompts with random or true values for $\mathbf{z}$ are referred to as filled prompts and answered prompts, respectively.

2.2. Methodology

This section introduces the event detection framework with joint dynamic adaptation and semantic relevance (DASR), which includes the following: (1) Prompt-based incremental learning: We leverage prompts to fine-tune pre-trained language models, generating incremental representations for newly introduced event types. (2) Trigger relevance estimation: We assess the semantic relevance between trigger words and event types to determine their importance. (3) Instance-guided prediction: We integrate the representations of event types and optimize both the incremental learning loss and classification loss to predict event types accurately. The DASR framework is illustrated in Figure 2.

2.2.1. Prompt-Based Incremental Tuning

Fine-tuning PLMs on new tasks carries the risk of losing information from previously learned data, a problem known as catastrophic forgetting. Class-incremental learning requires the model to retain knowledge of previously learned classes when new classes are added. We aim to use a more convenient prompt-based [40,41] fine-tuning paradigm to jointly assist in the dynamic updating of PLMs. First, the original definition of the prompt template is as follows:

$$f_p^E(x) = (x, E, y), \tag{1}$$

where x denotes the natural language describing the event, E denotes the event-related prompt word, and y denotes the event-type label. When we introduce incremental words IE (i.e., prompt words for new event types) in the prompt design, and aim to obtain prompts for all event types seen so far, the predefined prompt template can be further written as follows:

$$f_p^{IE}(x) = (x, IE, y), \tag{2}$$

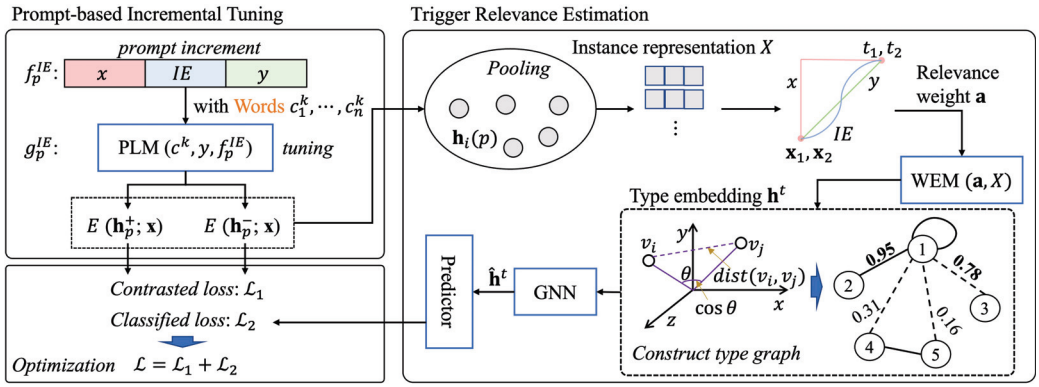


Figure 2. The framework of DASR. **Step 1**: New event types are incorporated into prompts and mapped to event representations jointly with PLM in contrastive training. **Step 2**: DASR captures the correlations of triggers in non-Euclidean space by modeling an event-type graph. **Step 3**: DASR is jointly optimized for the event detection loss.

For example, the initial template is designed for Sports Events, such as "{event_trigger} took place during the {event_type} at {location} on {date}", where the {event_type} refers to Sports Events, such as basketball games or football matches, and the trigger words are game or match. When a new event type, Political Protest is introduced, the template is automatically adjusted to the following: "{event_trigger} occurred during the {event_type} in {location} on {date}". In this case, the event type is Political Protest and the trigger word is protest.

We then adopt the replay strategy from initial incremental learning, utilizing PLMs to generate high-quality pseudo labels for new class events. For each event e, we obtain the words of the top k term frequency sequence from the corresponding sentence to initialize the prompt sentence. The prompts generated for the label words and the corresponding responses are as follows:

$$g_p^{IE}\left(y, c^k\right) = \left(y, c^k, x, IE, y\right), \tag{3}$$

where $c_i^k \in C$ represents the concept sequence of event e_i, which is used to evaluate the importance of words in the sentence. In short, DASR uses a two-step process to generate pseudo-labels for newly introduced event types. First, the model identifies candidate event triggers in unannotated data. Then, it applies the pre-trained PLM to generate high-confidence pseudo-labels for these events based on the identified triggers and context. These pseudo-labels are then used in incremental fine-tuning, allowing the model to adapt to new event categories without the need for manually labeled data. The process involves selecting the top-k most frequent event triggers and using PLMs to predict the event type associated with these triggers. This allows the model to efficiently adapt to new event types as they emerge.

Subsequently, we uniformly adopt BERT as the textual encoder for both prompt templates f and g. Leveraging the transformer architecture, BERT [42] generates contextual text representations conditioned on the given prompt, thereby preserving rich textual information. To enhance the label localization for novel events during the prompt tuning process, we introduce a contrastive learning loss constraint to clarify decision boundaries. We construct positive and negative sample pairs randomly, aiming for incremental prompt learning to pull similar samples of the new events closer and push dissimilar samples further apart in the feature space (using cosine similarity to measure similarity $\phi(,)$),

thereby improving the effectiveness of PLMs in event relevance extraction. The loss function is as follows:

$$\mathcal{L}_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(\phi\left(\mathbf{h}_i, \mathbf{h}_i^+\right)/\tau\right)}{\sum_{j=1}^{N}\exp\left(\phi\left(\mathbf{h}_i, \mathbf{h}_j\right)/\tau\right)} \tag{4}$$

where N denotes the number of sample pairs, $\mathbf{h}(f, g)$ denotes the sample representation in the latent space, and τ represents the temperature coefficient. BERT is a representative and widely used model in the current NLP field. We use BiLSTM and other models similar to BERT as benchmarks in our comparison experiments to ensure that the experimental results are representative. More complex pre-trained models (e.g., GPT series) require higher computational resources and longer training times, which are not conducive to quickly verifying the effectiveness of DASR.

2.2.2. Trigger Relevance Estimation

It is widely acknowledged that event types benefit the model in learning the semantic correlations between instances. Therefore, we combine event types with prompt embeddings to model a type-related graph structure. Then, leveraging the message-passing mechanism of GNNs, we capture type correlations to represent each event type as prior knowledge for instance type prediction.

For a sentence with the prompt phrase p and the candidate trigger m (including incorrect candidate instances), we input p into BERT to obtain the hidden representation $\mathbf{h}$ as prompt vectors (this is a simplified description of Section 2.2.1). Then, a pooling operation is applied to read out the representation for each candidate instance as a candidate trigger:

$$\mathbf{x}_i = \sigma(W\text{MAXPOOLING}\mathbf{h}_i(p)) \tag{5}$$

where W denotes the learnable parameter and σ denotes the activation function. In particular, a sentence instance is denoted as $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]$.

Relevance representation. Events are semantically composed of words, estimating the importance of words within events helps us quantify the underlying patterns of events. We introduce integrated gradients [43] to evaluate the importance of input features to the model's predictions. For the input vector $x \in \mathbb{R}^n$ and the model $\mathcal{F} : \mathbb{R}^n \to [0, 1]$, the attribution value for the i-th dimension is as follows:

$$\mathbf{a}_i = (\mathbf{x}_i - \mathbf{x}_i') \times \int_{\alpha=0}^{1}\frac{\partial\mathcal{F}(\mathbf{x}' + \alpha \times (\mathbf{x} - \mathbf{x}'))}{\partial\mathbf{x}_i}d\alpha, \tag{6}$$

where $\mathbf{x}'$ is an all-zero vector [44] used to align the input range, and the right term represents the gradient of $\mathcal{F}(\mathbf{x}_i)$ for $\mathbf{x}_i$. We then normalize Equation (6) to represent globally reasonable weights:

$$\mathbf{a}_i = \frac{\exp^{\|\mathbf{a}_i\|_2}}{\sum_{n=1}^{N}\exp^{\|\mathbf{a}_n\|_2}} \tag{7}$$

where $\|\cdot\|$ denotes the L_2 norm. Note that our goal always focuses on uncovering the semantic information provided by event types. Therefore, based on word relevance, for a specific event type, $T \in \mathcal{T}$, the type embedding is represented as follows:

$$\mathbf{h}^t = \frac{1}{|T|}\sum_{i=1}^{m}\mathbf{a}_i^T \times WEM(X) \tag{8}$$

where $|T|$ represents the event number of type. The left term of the multiplication denotes the learnable weights with semantic awareness, while the right term represents the event type with textual description encoded by the word-embedding model $WEM(\cdot)$ [45].

Type graph construction. As the initialization of edges in the type graph, we treat the types as nodes in the graph and calculate the possibility of edges existing between nodes using cosine similarity $\phi(,)$:

$$e_{ij} = \begin{cases} \phi\left(\mathbf{h}_i^t, \mathbf{h}_j^t\right), & \text{if} \phi\left(\mathbf{h}_i^t, \mathbf{h}_j^t\right) > \gamma \\ -\text{inf}, & \text{otherwise} \end{cases} \tag{9}$$

where γ is a hyperparameter threshold that satisfies the γ-neighbor strategy to select the most likely edges with the highest similarity scores, thus controlling the scale of the graph during construction.

Type representation. By modeling the graph structure, we can leverage the powerful relational representation capabilities of GNNs to further refine type representations. Specifically, after normalizing $\mathbf{E} = \left\{e_{ij} | i, j \in \{1, \ldots, n\}\right\}$ as an adjacency matrix, we learn node representations using a graph attention network [46]:

$$\mathbf{z}_i^{(l,k)} \leftarrow \sum_{j \in N_i \cup i} (\delta_{ij}^k \mathbf{E}) \mathbf{v}_j^{(l-1)}, \tag{10}$$

where δ_{ij}^k denotes normalized attention coefficients computed by the k-th attention mechanism, and $\mathbf{v}$ denotes randomly initialized embeddings. For the transformation, the k intermediate representations corresponding to K attention mechanisms are concatenated after the aggregation through a non-linear transformation to obtain the representation $\hat{\mathbf{h}}$ at layer l as follows:

$$\hat{\mathbf{h}}_i^{(l,t)} = \|_{k=1}^K \sigma\left(\mathbf{z}_i^{(l,k)} \mathbf{W}^{(k,l)}\right), \tag{11}$$

where $||$ denotes the concatenation operator and $\mathbf{W}$ denotes the learnable weight matrix. Thus, after message passing, we acquire informational knowledge from the neighbors associated with specific types, improving the type embeddings. The attention mechanism quantitatively enhances semantic relevance extraction by assigning higher importance to event types that are more semantically relevant to the current task, improving long-range dependency modeling. The final event representation is a weighted combination of these semantic dependencies, leading to more precise event detection.

2.2.3. Instance-Guided Prediction

In the previous design, we injected contextual instances with types into candidate triggers to obtain instance representations X and type representations $\hat{\mathbf{h}}^t$. Therefore, to disentangle the correlations in instances for downstream prediction tasks, we utilize token reconstruction based on contextual instances to predict the target instance types. This naturally covers the exploration of correlations between prompt sentences and semantic representations of instances.

Instance representation. To achieve robust type correlation extraction, we generate a set of handcrafted type representations $\hat{\mathbf{y}}$. Specifically, we evaluate the correlation by comparing the dot product scores between instance embeddings and type embeddings one by one, and after weighted aggregation, the probabilities of the generated predicted types are determined. This process is formally represented as follows:

$$\hat{\mathbf{y}}_i = \sum_{j=1}^{N+1} \left(\frac{\exp\left(\mathbf{x}_i \cdot \hat{\mathbf{h}}_j^t\right)}{\sum_{k=1}^{N+1} \left(\mathbf{x}_i \cdot \hat{\mathbf{h}}_k^t\right)} \hat{\mathbf{h}}_j^t\right), \tag{12}$$

where $N+1$ represents the number of types, indicating that the reconstructed instance type t_k is introduced. Next, we generate sequential handcrafted type representations to predict

the target instance type based on the context. For the target instance x_i, we replace its type representation with $\mathbf{e}_{mask}$, and the corresponding sequential instance type representation is as follows:

$$\hat{\mathbf{Y}}^{(i)} = [\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_{i-1}, \mathbf{e}_{\text{mask}}, \hat{\mathbf{y}}_{i+1}, \ldots, \hat{\mathbf{y}}_m]. \tag{13}$$

To generate more correlated target type representations when fusing instance type representations with instance text representations, we employ a neural network for linear fusion, allowing the candidate instances to be re-represented in a learnable manner:

$$\mathbf{Z}^{(i)} = \sigma\Big(\hat{\mathbf{W}}\Big[\hat{\mathbf{Y}}^{(i)}; X\Big]\Big), \tag{14}$$

where $\hat{\mathbf{W}}$ denotes the linear fusion parameter and $[;]$ denotes the element-wise vector concatenation.

Type prediction. To identify the handcrafted event types, t_k, and the real event types, t_j, within the candidate instances, we obtain the posterior probabilities in the final predictor as follows:

$$p(\hat{\mathbf{h}}^t|\mathbf{x}_i) = \text{SOFTMAX}(\hat{\mathbf{h}}^t \cdot \hat{\mathbf{z}}_i). \tag{15}$$

This design provides meaningful guidance for event detection. Specifically, when the target type is t_k, the model predicts a higher $p(\hat{\mathbf{h}}_k^t|\mathbf{x}_i)$ to improve generalization ability when encountering new types. Conversely, when the target type is a known event type, the model no longer distinguishes precisely which specific type it is, instead exhibiting a uniformly lower $p(\hat{\mathbf{h}}_k^t|\mathbf{x}_i)$. Overall, the final prediction result is the type with the highest posterior probability. We define the binary cross-entropy loss for type relevance learning as follows:

$$\begin{aligned}\mathcal{L}_2 = &- \sum_{\mathbf{x}_i \in X} \sum_{t_j \in \mathcal{T}} \mathbf{y}_{ij} \log p\Big(\hat{\mathbf{h}}_j^t \mid \mathbf{x}_i\Big) \\ &+ (1 - \mathbf{y}_{ij}) \log\Big(1 - p\Big(\hat{\mathbf{h}}_j^t \mid \mathbf{x}_i\Big)\Big)\end{aligned} \tag{16}$$

Our overall model optimization objective is as follows:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_2 \tag{17}$$

where λ is a balancing coefficient used to weigh the impact of incremental learning and type representation on model performance.

3. Results

This section conducts experiments to evaluate the performance of DASR. The experimental results demonstrate that the proposed framework effectively mitigates catastrophic forgetting and significantly improves the accuracy and adaptability of event detection when dealing with evolving data distributions and newly introduced event types.

3.1. Experimental Setup

Datasets. Our experiment was conducted on the publicly available, large-scale dataset, i.e., the Twitter dataset collected by Twitter API for incremental event detection, and the MAVEN and ACE-2005 datasets for offline event detection to evaluate the effectiveness of DASR.

The Twitter dataset, after cleaning out repeated, still contains 68,841 manually labeled tweets related to 503 event classes distributed over a period of approximately 4 weeks (29 days). To simulate evolving data distributions, the Twitter dataset is divided into sequential message blocks by date. Each block introduces new event types, ensuring that the model is exposed to a continuously changing environment. In our experiments, we split the Twitter dataset by date. Specifically, we use the messages of the first week to form

an initial message block, M_0, and the messages of the remaining days to form the following message blocks, $M_1, M_2, \ldots, M_{21}$. Furthermore, MAVEN is a newly constructed large-scale fine-grained corpus, which contains 4480 documents with 168 fine-grained event types. We also use the similar ACE-2005 dataset for evaluation, which contains 599 documents with 33 event types.

Baselines. To demonstrate the significant performance of our model, we select the following baselines. **BERT** [42], which is based on the Transformer architecture and deeply represents natural language through pre-training and finetuning; **BiLSTM** [47], which learns long-term bidirectional semantic dependencies; **EventX** [48], a fine-grained event detection method based on community detection suitable for online scenarios; **EE-GCN** [49], which simultaneously exploits syntactic structure and typed dependency label information to perform ED; **KPGNN** [25], a knowledge-preserving incremental heterogeneous graph neural network model for streaming social event detection.

Settings. We set the number of attention heads for the relation aggregation GNN to 8, the embedding dimension to 64, the total number of layers to 2, the learning rate to 0.001, and the optimizer to Adam. The training spans 100 epochs with a patience of 5 for early stopping. The dimension of the $WEM(\cdot)$ embedding is set to 300, the threshold γ is 0.4, λ is 0.6, and the dropout rate is 0.5. We select the best model for inference. BiLSTM and other GNN-based baselines generally use the same configuration, except that the dimension is set to 32. For EventX, we adopt the hyperparameters as suggested in the original paper [48]. We repeat all experiments five times and report the mean and standard variance of the results.

Evaluation metrics. For incremental event detection, to evaluate the performance of all models, we use adjusted mutual information (AMI) [50] and adjusted rand index (ARI) [50] to measure the similarity between ground-truth clusters and their detected message clusters. AMI measures the mutual information between two clusters and adjusts it based on chance. ARI considers all predicted label pairs and chance and calculates pairs allocated in the same or different clusters. Adaptation and mitigation of the catastrophic forgetting problem are assessed by comparing performance on new event types introduced in incremental learning scenarios while ensuring that the performance on previously learned types remains stable. For offline event detection, we employ a comprehensive set of evaluation metrics including precision (P), which measures the accuracy of positive predictions; recall (R), which evaluates the model's ability to identify all relevant instances; and the F1 score, which provides a balanced measure of both precision and recall. The metrics used follow the setup of most of the existing research.

3.2. Performance Comparison

Incremental evaluation. We evaluate DASR in the incremental event detection scenario. As shown in Table 1, we summarize the performances of several state-of-the-art methods in terms of AMI and ARI metrics. The results demonstrate stable performance as new event types emerge, with a peak AMI of 0.80 and ARI of 0.77 on block M6, where the diversity of event types is the highest. The GNN-based methods have achieved advantages across different message blocks, with DASR showing the most significant improvement. Compared to the best-performing model in the baseline, DASR improved the average AMI by 4.3% and the average ARI by 4.0%, with the best results showing a 27% increase in AMI for M1 and a 20% increase in ARI for M15. This indicates that our design effectively retains the most recent knowledge during incremental training and better mitigates the catastrophic forgetting issue as messages accumulate. In contrast, EE-GCN and KPGNN might sometimes be distracted by outdated information. The generally low performances of BERT and BiLSTM are attributed to their neglect of the semantic relevance of events in

non-Euclidean space, while EventX's performance in incremental scenarios is limited due to its sole reliance on community detection. In addition, DRSA performs poorly at M19 due to fewer instances of certain event types in the training data, and the model may not be able to adequately learn the features of these types, resulting in poor performance.

Offline evaluation. To further demonstrate the advantages of DASR, we analyze the results of all methods on two offline datasets across three metrics, as shown in Table 2. We follow the standard data split method used in GNNs [34], randomly selecting 70% for training, 10% for validation, and 20% for testing. DASR achieves the top rankings among all methods, with an average improvement of 2.1% across all metrics, and a maximum improvement of 6.2%. This indicates that our approach maintains excellent performance on offline data as well. The additional prompt design does not cause classification bias, and the deep semantic information provides the model with higher-quality information.

3.3. Further Analysis

In this section,we perform further ablation studies, visualization, and hyperparameter analysis of DASR.

Ablation study. As shown in Figure 3, we conducted ablation experiments on the dynamic adjustment (w/o prompt) and semantic relevance (w/o relevance) components of DASR separately to illustrate the effectiveness of each component. Figure 3a,b present the AMI and ARI performances after removing the prompt increment method from DASR. DASR exhibits relatively stable performance across most blocks, with a significant AMI advantage in the 5th, 10th, and 15th blocks. EventX performs significantly lower, with ARI remaining below 0.2 in almost all blocks. EE-GCN and KPGNN show similar performance, generally matching or slightly below that of DASR. Figure 3c,d show the AMI and ARI performances after removing the semantic relevance in DASR. DASR demonstrated high levels and stability, significantly outperforming competing methods in multiple blocks. Although BERT and BiLSTM did not perform as well as DASR, BERT maintained a secondary optimal performance in several blocks, while BiLSTM showed weaker and more fluctuating performance.

Overall, prompt-based incremental learning successfully conveys semantic information across event types by introducing dynamic prompt templates, while avoiding the destruction of existing knowledge by full model retraining. The ablation results show that the module is indispensable for handling dynamic data distributions and new event types. Graph-based semantic relevance modeling effectively captures the complex dependencies between trigger words and event types through the structure of event-type graphs in non-Euclidean space. This module plays a crucial role in improving semantic representation and enhancing overall performance.

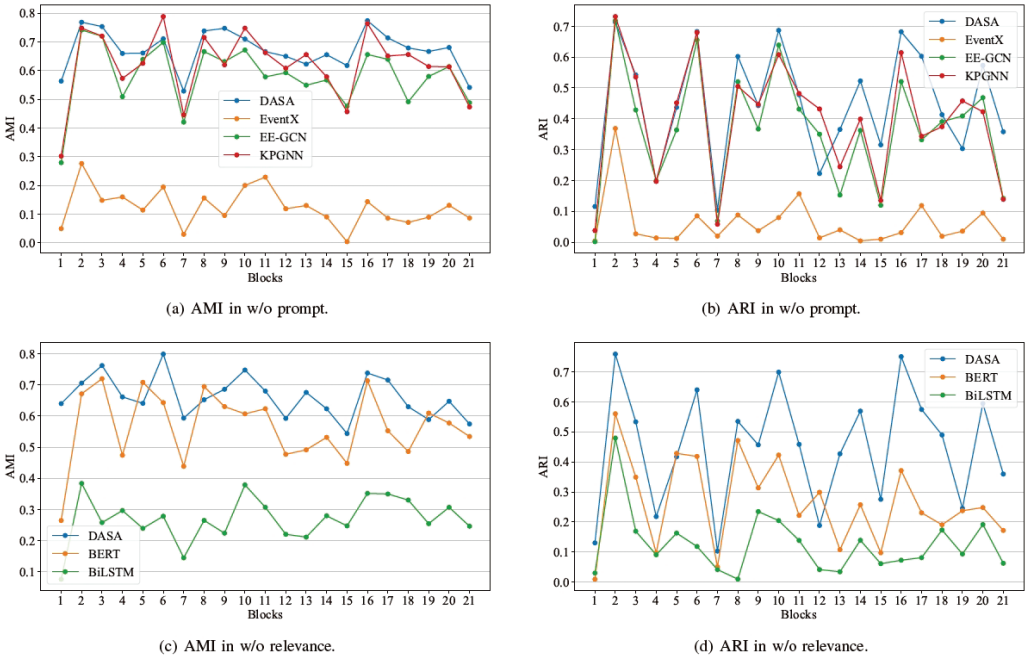
Visualization. We visualize the model's predictive outputs, leveraging t-SNE [51] for dimensionality reduction. Figure 4 illustrates the differentiation of data points in the M_3 block using colors to distinguish between different classes. It can be observed that DASR demonstrates the best performance, with its attention mechanism and incremental learning capabilities contributing to exceptional semantic representation and class distinction, resulting in clear clustering boundaries. EE-GCN and KPGNN follow as the next best performers. These models, through graph networks and their enhancement methods, effectively capture semantic relationships, resulting in compact clustering with clear class boundaries. BERT shows moderate performance. It generates relatively compact clusters through the bi-directional representation of the Transformer architecture but still exhibits overlap between categories. BiLSTM and EventX perform poorly. BiLSTM relies on word order information but is weak in semantic clustering, while EventX has the worst clustering effect. In summary, the DASR method exhibits high clarity and separability, further highlighting the method's classification advantages.

Table 1. Summary of incremental evaluation AMI and ARI on the Twitter dataset. The best results are marked in bold. the mean and standard deviation of results are reported with 5 trials.

Method	BERT		BiLSTM		EventX		EE-GCN		KPGNN		DASR	
Metric	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI
M_1	0.34 ± 0.00	0.03 ± 0.00	0.12 ± 0.00	0.03 ± 0.00	0.06 ± 0.00	0.01 ± 0.00	0.35 ± 0.01	0.05 ± 0.00	0.37 ± 0.00	0.07 ± 0.01	**0.64 ± 0.01**	**0.20 ± 0.00**
M_2	0.76 ± 0.00	0.64 ± 0.00	0.41 ± 0.00	0.49 ± 0.00	0.29 ± 0.02	0.45 ± 0.02	0.77 ± 0.00	0.75 ± 0.00	0.78 ± 0.01	0.76 ± 0.02	**0.80 ± 0.00**	**0.77 ± 0.02**
M_3	0.73 ± 0.00	0.43 ± 0.00	0.31 ± 0.00	0.17 ± 0.00	0.18 ± 0.01	0.09 ± 0.01	0.73 ± 0.00	0.48 ± 0.02	0.74 ± 0.00	0.58 ± 0.01	**0.79 ± 0.00**	**0.59 ± 0.00**
M_4	0.55 ± 0.00	0.19 ± 0.00	0.30 ± 0.00	0.11 ± 0.00	0.19 ± 0.01	0.07 ± 0.01	0.60 ± 0.00	0.23 ± 0.00	0.64 ± 0.01	**0.29 ± 0.01**	**0.68 ± 0.00**	0.26 ± 0.03
M_5	0.71 ± 0.00	0.44 ± 0.00	0.33 ± 0.00	0.19 ± 0.00	0.14 ± 0.00	0.04 ± 0.00	0.71 ± 0.01	0.45 ± 0.00	**0.71 ± 0.01**	0.47 ± 0.03	0.71 ± 0.00	**0.49 ± 0.00**
M_6	0.74 ± 0.00	0.44 ± 0.00	0.36 ± 0.00	0.18 ± 0.00	0.27 ± 00	0.14 ± 0.00	0.76 ± 0.00	**0.74 ± 0.02**	0.79 ± 0.01	0.72 ± 0.03	**0.81 ± 0.02**	0.73 ± 0.01
M_7	0.50 ± 0.00	0.07 ± 0.00	0.20 ± 0.00	0.08 ± 0.00	0.13 ± 0.00	0.02 ± 0.00	0.50 ± 0.02	0.10 ± 0.01	0.51 ± 0.01	0.12 ± 0.00	**0.62 ± 0.00**	**0.18 ± 0.00**
M_8	0.75 ± 0.00	0.50 ± 0.00	0.35 ± 0.00	0.08 ± 0.00	0.21 ± 0.00	0.09 ± 0.00	0.75 ± 0.02	0.58 ± 0.01	**0.76 ± 0.01**	0.60 ± 0.01	0.75 ± 0.01	**0.62 ± 0.00**
M_9	0.66 ± 0.00	0.33 ± 0.00	0.32 ± 0.00	0.27 ± 0.00	0.19 ± 0.00	0.07 ± 0.00	0.68 ± 0.00	0.41 ± 0.03	0.71 ± 0.02	**0.46 ± 0.02**	**0.75 ± 0.01**	0.46 ± 0.01
M_{10}	0.70 ± 0.00	0.44 ± 0.00	0.39 ± 0.00	0.22 ± 0.00	0.24 ± 0.00	0.13 ± 0.00	0.76 ± 0.00	**0.73 ± 0.05**	0.78 ± 0.01	0.70 ± 0.06	**0.80 ± 0.02**	0.73 ± 0.03
M_{11}	0.65 ± 0.00	0.27 ± 0.00	0.37 ± 0.00	0.17 ± 0.00	0.24 ± 0.00	0.16 ± 0.00	0.67 ± 0.02	0.45 ± 0.05	0.71 ± 0.01	0.49 ± 0.03	**0.72 ± 0.01**	**0.51 ± 0.06**
M_{12}	0.56 ± 0.00	0.31 ± 0.00	0.32 ± 0.00	0.13 ± 0.00	0.16 ± 0.00	0.07 ± 0.00	0.62 ± 0.01	0.42 ± 0.03	0.66 ± 0.01	**0.48 ± 0.01**	**0.67 ± 0.00**	0.45 ± 0.00
M_{13}	0.59 ± 0.00	0.14 ± 0.00	0.31 ± 0.00	0.13 ± 0.00	0.16 ± 0.00	0.04 ± 0.00	0.61 ± 0.02	0.25 ± 0.01	0.67 ± 0.01	**0.29 ± 0.03**	**0.69 ± 0.01**	0.23 ± 0.00
M_{14}	0.61 ± 0.00	0.30 ± 0.00	0.34 ± 0.00	0.16 ± 0.00	0.14 ± 00	0.10 ± 0.00	0.62 ± 0.02	0.38 ± 0.01	0.65 ± 0.00	0.42 ± 0.02	**0.72 ± 0.02**	**0.62 ± 0.03**
M_{15}	0.50 ± 0.00	0.10 ± 0.00	0.26 ± 0.00	0.14 ± 0.00	0.07 ± 0.00	0.01 ± 0.00	0.53 ± 0.00	0.14 ± 0.03	0.54 ± 0.00	0.17 ± 0.00	**0.63 ± 0.01**	**0.37 ± 0.02**
M_{16}	0.72 ± 0.00	0.41 ± 0.00	0.41 ± 0.00	0.10 ± 0.00	0.19 ± 0.00	0.08 ± 0.00	0.75 ± 0.00	0.59 ± 0.06	0.77 ± 0.01	0.66 ± 0.05	**0.80 ± 0.02**	**0.78 ± 0.01**
M_{17}	0.60 ± 0.00	0.24 ± 0.00	0.35 ± 0.00	0.17 ± 0.00	0.18 ± 0.00	0.12 ± 0.00	0.67 ± 0.00	0.41 ± 0.00	0.68 ± 0.01	0.43 ± 0.05	**0.72 ± 0.01**	**0.51 ± 0.01**
M_{18}	0.53 ± 0.00	0.24 ± 0.00	0.35 ± 0.00	0.19 ± 0.00	0.16 ± 00	0.08 ± 0.00	0.59 ± 0.02	0.44 ± 0.00	0.66 ± 0.02	0.47 ± 0.04	**0.70 ± 0.01**	**0.49 ± 0.01**
M_{19}	0.63 ± 0.00	0.32 ± 0.00	0.35 ± 0.00	0.16 ± 0.00	0.16 ± 0.00	0.07 ± 0.00	0.67 ± 0.01	0.47 ± 0.01	**0.71 ± 0.01**	**0.51 ± 0.03**	0.68 ± 0.00	0.33 ± 0.00
M_{20}	0.62 ± 0.00	0.33 ± 0.00	0.34 ± 0.00	0.20 ± 0.00	0.18 ± 0.00	0.11 ± 0.00	0.66 ± 0.00	0.49 ± 0.01	0.68 ± 0.02	0.51 ± 0.04	**0.70 ± 0.01**	**0.61± 0.02**
M_{21}	0.57 ± 0.00	0.18 ± 0.00	0.31 ± 0.00	0.16 ± 0.00	0.10 ± 0.00	0.01 ± 0.00	0.57 ± 0.00	0.19 ± 0.01	0.57 ± 0.00	0.20 ± 0.01	**0.61 ±0.00**	**0.38 ± 0.00**

Table 2. Results of event detection on the MAVEN and ACE-2005 datasets. The best results are marked in bold. The mean and standard deviation of the results are reported based on 5 trials.

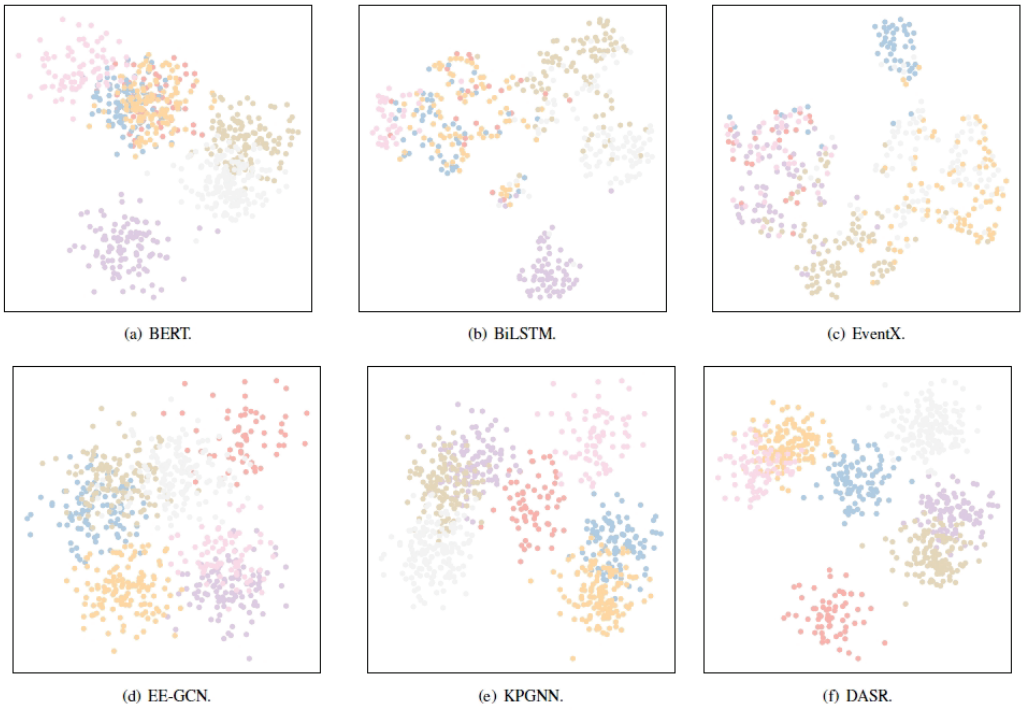
Dataset	MAVEN			ACE-2005			Avg. Rank
Metric	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	
BERT	65.0 ± 0.84	**70.9 ± 0.94**	67.8 ± 0.15	71.3 ± 1.77	77.1 ± 1.99	74.1 ± 1.56	3.8
BiLSTM	63.4 ± 0.70	64.8 ± 0.69	64.1 ± 0.13	77.2 ± 2.08	74.9 ± 2.62	75.4 ± 1.64	5.2
EventX	64.8 ± 0.91	68.5 ± 1.11	65.5 ± 0.27	74.6 ± 1.88	76.0 ± 2.00	74.9 ± 1.31	4.8
EE-GCN	64.3 ± 1.67	68.8 ± 1.38	66.9 ± 0.57	76.9 ± 1.72	78.5 ± 1.62	77.2 ± 1.43	3.5
KPGNN	67.0 ± 0.56	70.1 ± 1.08	68.2 ± 0.35	**77.8 ± 1.52**	77.6 ± 2.30	79.1 ± 1.24	2.0
DASR	**68.2 ± 0.51**	69.9 ± 0.85	**68.9 ± 0.66**	77.5 ± 1.30	78.4 ± 2.01	**79.7 ± 0.57**	1.7



(a) AMI in w/o prompt. (b) ARI in w/o prompt.

(c) AMI in w/o relevance. (d) ARI in w/o relevance.

Figure 3. Ablation study on M_3 of the Twitter dataset.



(a) BERT. (b) BiLSTM. (c) EventX.

(d) EE-GCN. (e) KPGNN. (f) DASR.

Figure 4. Visualization on M_3 of the Twitter dataset.

Hyperparameter analysis. Figure 5 shows the impact of the hyperparameter γ on the performance of DASR, which represents the threshold for generating edges when constructing the type graph in DASR. As γ increases, the number of edges in the graph grows. The figure includes two curves representing the AMI and ARI scores at different γ values, along with their fitting curves (Fit_AMI and Fit_ARI). From the figure, it can be seen that when γ is around 0.4, both AMI and ARI scores reach their peak, indicating that the graph structure generated at this threshold is the most beneficial for improving the model's performance. As γ continues to increase, the AMI and ARI scores gradually decline, possibly because excessive edges in the network introduce noise, affecting

the model's learning efficacy. Figure 6 shows the impact of the hyperparameter γ on the performance of DASR, which represents the trade-off in the loss function between incremental learning and the relevant neural network learning components during DASR training. A larger γ indicates greater emphasis on relevance learning, while a smaller γ emphasizes incremental learning. The figure shows that when γ is around 0.6, the AMI score is the highest, whereas the ARI score reaches a relatively high level when γ is 0.3. Performance is poorer when γ is at an extreme value (0 or 1), suggesting that relying solely on either incremental learning or relevance learning is less ideal. Finding a balance between the two optimizes the model's performance. Overall, by adjusting the hyperparameters γ and γ, the performance of DASR can be effectively enhanced, allowing it to better adapt to different tasks and datasets.

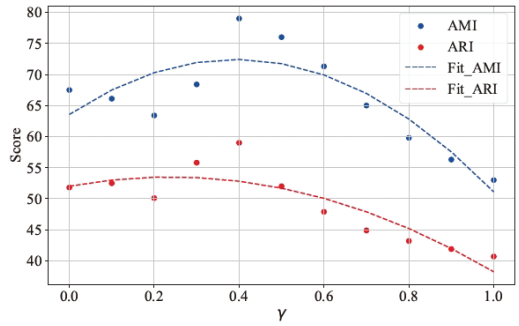


Figure 5. Hyperparameter γ analysis on M_3 of the Twitter dataset.

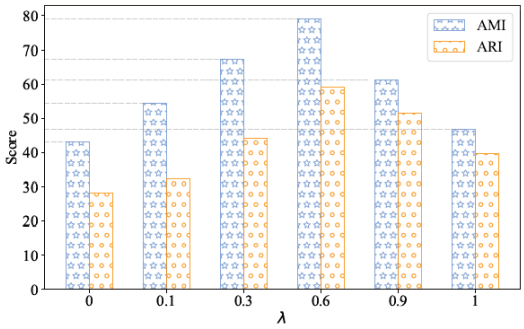


Figure 6. Hyperparameter λ analysis on M_3 of the Twitter dataset.

4. Conclusions

In this paper, we propose an innovative event detection framework (DASR), which cleverly integrates dynamic event adaptation with semantic relevance mining to address the problem of catastrophic forgetting while improving the adaptability and accuracy of the model in event detection. The core components of our approach include leveraging a pre-trained language model for extensive context understanding, integrating an incremental learning module with prompts to model the temporal sequence of events, and utilizing multi-perspective correlations along with a self-attention mechanism to handle entity recognition and event trigger word identification. Extensive experiments validate the efficacy of DASR, demonstrating its outstanding performance in mitigating catastrophic forgetting and adapting to new event types, thereby significantly advancing the state-of-the-art in event detection tasks.

Author Contributions: Conceptualization and design of the study, X.Z., G.L. and K.Q.; Methodology, X.Z. and G.L.; Data curation, X.Z.; formal analysis, X.Z. and K.Q.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z., G.L. and K.Q.; investigation, X.Z.; Project administration, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data presented in this work can be obtained from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest with the institutions, organizations, or companies mentioned in the manuscript.

References

1. Nguyen, T.M.; Nguyen, T.H. One for all: Neural joint modeling of entities and events. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 6851–6858.
2. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimed. Tools Appl.* **2023**, *82*, 3713–3744. [CrossRef] [PubMed]
3. Atefeh, F.; Khreich, W. A survey of techniques for event detection in twitter. *Comput. Intell.* **2015**, *31*, 132–164. [CrossRef]
4. Lewis, D.D.; Jones, K.S. Natural language processing for information retrieval. *Commun. ACM* **1996**, *39*, 92–101. [CrossRef]
5. Hirschman, L.; Gaizauskas, R. Natural language question answering: The view from here. *Nat. Lang. Eng.* **2001**, *7*, 275–300. [CrossRef]
6. Tien, J.M. Internet of things, real-time decision making, and artificial intelligence. *Ann. Data Sci.* **2017**, *4*, 149–178. [CrossRef]
7. Gan, C.; Wang, N.; Yang, Y.; Yeung, D.-Y.; Hauptmann, A.G. Devnet: A deep event network for multimedia event detection and evidence recounting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2568–2577.
8. Liu, J.; Chen, Y.; Liu, K.; Zhao, J. Event detection via gated multilingual attention mechanism. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
9. Zhao, L.; Sun, Q.; Ye, J.; Chen, F.; Lu, C.-T.; Ramakrishnan, N. Multi-task learning for spatio-temporal event forecasting. In Proceedings of the 21th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1503–1512.
10. Allan, J.; Papka, R.; Lavrenko, V. On-line new event detection and tracking. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 24–28 August 1998; pp. 37–45.
11. Sheng, J.; Sun, R.; Guo, S.; Cui, S.; Cao, J.; Wang, L.; Liu, T.; Xu, H. CorED: Incorporating type-level and instance-level correlations for fine-grained event detection. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1122–1132.
12. Wang, S.; Yu, M.; Huang, L. The art of prompting: Event detection based on type specific prompts. *arXiv*, **2022**, arXiv:2204.07241.
13. Yuan, H.; Sun, Q.; Fu, X.; Ji, C.; Li, J. Dynamic Graph Information Bottleneck. In Proceedings of the ACM on Web Conference 2024, Singapore, 13–17 May 2024; pp. 469–480.
14. Wei, Y.; Yuan, H.; Fu, X.; Sun, Q.; Peng, H.; Li, X.; Hu, C. Poincaré Differential Privacy for Hierarchy-aware Graph Embedding. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; pp. 9160–9168.
15. Yang, Z.; Wei, Y.; Li, H.; Li, Q.; Jiang, L.; Sun, L.; Yu, X.; Hu, C.; Peng, H. Adaptive Differentially Private Structural Entropy Minimization for Unsupervised Social Event Detection. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, Boise, ID, USA, 21–25 October 2024; pp. 2950–2960.
16. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
17. Hayes, T.L.; Kafle, K.; Shrestha, R.; Acharya, M.; Kanan, C. Remind your neural network to prevent catastrophic forgetting. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 466–483.
18. Van de Ven, G.M.; Tuytelaars, T.; Tolias, A.S. Three types of incremental learning. *Nat. Mach. Intell.* **2022**, *4*, 1185–1197. [CrossRef]
19. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **2019**, *113*, 54–71. [CrossRef] [PubMed]
20. Wu, T.; Caccia, M.; Li, Z.; Li, Y.F.; Qi, G.; Haffari, G. Pretrained language model in continual learning: A comparative study. In Proceedings of the International Conference on Learning Representations 2022, Virtual Event, 25–29 April 2022.
21. Biesialska, M.; Biesialska, K.; Costa-Jussa, M.R. Continual lifelong learning in natural language processing: A survey. *arXiv* **2020**, arXiv:2012.09823.
22. Yu, P.; Ji, H.; Natarajan, P. Lifelong event detection with knowledge transfer. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 5278–5290.
23. Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; Pfister, T. Learning to prompt for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 139–149.

24. Cao, P.; Chen, Y.; Zhao, J.; Wang, T. Incremental event detection via knowledge consolidation networks. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 707–717.
25. Cao, Y.; Peng, H.; Wu, J.; Dou, Y.; Li, J.; Yu, P.S. Knowledge-preserving incremental social event detection via heterogeneous gnns. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 12–23 April 2021; pp. 3383–3395.
26. Peng, H.; Zhang, R.; Li, S.; Cao, Y.; Pan, S.; Philip, S.Y. Reinforced, incremental and cross-lingual event detection from social messages. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 980–998. [CrossRef] [PubMed]
27. Li, P.; Yu, X.; Peng, H.; Xian, Y.; Wang, L.; Sun, L.; Zhang, J.; Yu, P.S. Relational Prompt-based Pre-trained Language Models for Social Event Detection. *ACM Trans. Inf. Syst.* **2024**, *43*, 1–43. [CrossRef]
28. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. Exploring pre-trained language models for event extraction and generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5284–5294.
29. Wang, Z.; Wang, X.; Han, X.; Lin, Y.; Hou, L.; Liu, Z.; Li, P.; Li, J.; Zhou, J. CLEVE: Contrastive pre-training for event extraction. *arXiv* **2021**, arXiv:2105.14485.
30. Araki, J.; Mitamura, T. Joint event trigger identification and event coreference resolution with structured perceptron. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2074–2080.
31. Li, Q.; Ji, H.; Huang, L. Joint event extraction via structured prediction with global features. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; Volume 1: Long Papers, pp. 73–82.
32. Nguyen, T.H.; Grishman, R. Event detection and domain adaptation with convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 2: Short Papers, pp. 365–371.
33. Feng, X.; Qin, B.; Liu, T. A language-independent neural network for event detection. *Sci. China Inf. Sci.* **2018**, *61*, 092106. [CrossRef]
34. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
35. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
36. Zhang, Y.; Li, Z.; Liu, Z.; Zheng, H.-T.; Shen, Y.; Zhou, L. Event detection with dynamic word-trigger-argument graph neural networks. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3858–3869. [CrossRef]
37. Nguyen, T.; Grishman, R. Graph convolutional networks with argument-aware pooling for event detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
38. Fu, X.; Wei, Y.; Sun, Q.; Yuan, H.; Wu, J.; Peng, H.; Li, J. Hyperbolic geometric graph representation learning for hierarchy-imbalance node classification. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–4 May 2023; pp. 460–468.
39. Sun, Q.; Li, J.; Peng, H.; Wu, J.; Fu, X.; Ji, C.; Philip, S.Y. Graph structure learning with variational information bottleneck. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 28 February–1 March 2022; pp. 4165–4174.
40. Varshney, V.; Patidar, M.; Kumar, R.; Shroff, G.; Vig, L. Prompt augmented generative replay via supervised contrastive training for lifelong intent detection. In *Findings of the Association for Computational Linguistics: NAACL 2022*; Association for Computational Linguistics: Seattle, WC, USA, 2024.
41. Zhang, S.; Ji, T.; Ji, W.; Wang, X. Zero-shot event detection based on ordered contrastive learning and prompt-based prediction. In *Findings of the Association for Computational Linguistics: NAACL 2022*; Association for Computational Linguistics: Seattle, WC, USA, 2022; pp. 2572–2580.
42. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
43. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017, pp. 3319–3328.
44. Wallace, E.; Tuyls, J.; Wang, J.; Subramanian, S.; Gardner, M.; Singh, S. AllenNLP interpret: A framework for explaining predictions of NLP models. *arXiv* **2019**, arXiv:1909.09251.
45. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
46. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
47. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef] [PubMed]
48. Liu, B.; Han, F.X.; Niu, D.; Kong, L.; Lai, K.; Xu, Y. Story forest: Extracting events and telling stories from breaking news. *ACM Trans. Knowl. Discov. Data (TKDD)* **2020**, *14*, 31. [CrossRef]
49. Cui, S.; Yu, B.; Liu, T.; Zhang, Z.; Wang, X.; Shi, J. Edge-enhanced graph convolution networks for event detection with syntactic relation. *arXiv* **2020**, arXiv:2002.10757.

50. Vinh, N.; Epps, J.; Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.* **2010**, *11*, 2837–2854.
51. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

electronics

MDPI

Article

Hardness Classification Using Cost-Effective Off-the-Shelf Tactile Sensors Inspired by Mechanoreceptors

Yash Sharma *, Pedro Ferreira and Laura Justham

Wolfson School of Mechanical, Electrical, and Manufacturing Engineering, Loughborough University, Loughborough LE11 3TU, UK; p.ferreira@lboro.ac.uk (P.F.); l.justham@lboro.ac.uk (L.J.)
* Correspondence: y.sharma@lboro.ac.uk

Abstract: Perception is essential for robotic systems, enabling effective interaction with their surroundings through actions such as grasping and touching. Traditionally, this has relied on integrating various sensor systems, including tactile sensors, cameras, and acoustic sensors. This study leverages commercially available tactile sensors for hardness classification, drawing inspiration from the functionality of human mechanoreceptors in recognizing complex object properties during grasping tasks. Unlike previous research using customized sensors, this study focuses on cost-effective, easy-to-install, and readily deployable sensors. The approach employs a qualitative method, using Shore hardness taxonomy to select objects and evaluate the performance of commercial off-the-shelf (COTS) sensors. The analysis includes data from both individual sensors and their combinations analysed using multiple machine learning approaches, and accuracy as the primary evaluation metric was considered. The findings illustrate that increasing the number of classification classes impacts accuracy, achieving 92% in binary classification, 82% in ternary, and 80% in quaternary scenarios. Notably, the performance of commercially available tactile sensors is comparable to those reported in the literature, which range from 50% to 98% accuracy, achieving 92% accuracy with a limited data set. These results highlight the capability of COTS tactile sensors in hardness classification giving accuracy levels of 92%, while being cost-effective and easier to deploy than customized tactile sensors.

Keywords: hardness classification; COTS tactile sensor; Shore hardness scale; mechanoreceptors

1. Introduction

In the robotics domain, various sensor systems, including tactile sensors, cameras, and acoustic sensors, have been utilized to provide perceptual capabilities. These sensors are often integrated into gripper systems, equipped as artificial fingers, layered sensors, or embedded modules. Previous research has explored the use of different sensors for several classifications [1], using sensors individually and in combination. Material classification can be performed to determine the texture of the object or the hardness of the object based on some scale [1,2], whereas object classification studies have involved grasping different objects to analyse their characteristics using sensor values and different resistance values. Techniques such as pressing, sliding sensors across surfaces, and employing squeezing or grasping have been explored in the literature as state-of-the-art methodologies in tactile perception [1,2]. However, the selection of sensors in these studies has typically been driven by application-specific requirements rather than mirroring the architecture or functionality of human mechanoreceptors [3–5]. In humans, tactile information is processed by four distinct mechanoreceptors, each specializing in detecting different tactile signals [6]. Presently, there is a gap in the field concerning the selective integration of sensors which could have the same functionality of human mechanoreceptors to perform hardness classification.

The aim of this study is to explore commercially available tactile sensors that mimic the properties and functionalities of human mechanoreceptors. This research seeks to identify sensors capable of detecting tactile information with a performance comparable to that of human fingers, emphasizing cost-effectiveness. Previous research has primarily focused on utilizing customized tactile sensors that are expensive and application-specific [7–13]. The specialized architecture of these sensors limits their immediate deployment. This work targets the identification of tactile sensors that are readily deployable, requiring minimal installation efforts and offering accessibility without the need for extensive customization. It is important to establish a framework that allows for the description of tactile sensor selection, drawing from the mechanoreceptor concepts described in the existing literature [6,14]. Although prior studies have proposed advanced tactile sensors as artificial mechanoreceptors [5,15–21], their development and implementation in robotics systems are still at an early stage. In contrast, the existing array of tactile sensors in robotics encompasses various modalities, including vibration, thermal, piezoelectric, and piezoresistive sensors [1,22], which collectively provide a rich source of tactile data. However, many of these sensors are customized for specific tasks or applications, and they have not been adequately assessed for classifying materials based on hardness for robotic grasping tasks. Therefore, there is a need to investigate the potential of commercial or cost-effective tactile sensors in hardness classification in robotic manipulation scenarios. Sensors in robotics grippers [8–10,23–25] and receptors in human hands [6,14,26–28] play a crucial role in receiving tactile information, forming the foundation of tactile perception systems in both. In robotics, hardness classification utilizes various tactile sensing methods, where measuring mechanical resistance while squeezing an object is crucial [1,8,29]. Hardness classification judges an object's resistance to deformation, its ability to withstand external forces without distortion, and its tendency to deformation, thereby enabling classification into categories such as hard, soft, or flexible. While most research has primarily focused on binary classifications of hardness and softness [8,9,24,30–32], there remains a significant gap in exploring classifications that involve three or more classes. Developing multi-class systems could offer distinct advantages in real-time applications, particularly in robotic manipulation.

Tactile information obtained through grasping is crucial for understanding an object's tactile properties [8,33], unlike vision systems, which rely on predefined object features and lack predictive abilities where tactile sensors provide numeric data, enabling more feasible and less time-consuming analysis when coupled with machine learning algorithms [34]. This paper proposes the use of cost-effective commercial off-the-shelf tactile sensors, inspired by human mechanoreceptors, to perform hardness classification through grasping methods. Human tactile perception relies on the significant abilities of mechanoreceptors, which are specialized sensors within our bodies that detect pressure, vibration, and texture. These biological sensors enable us to recognize various tactile properties with notable precision. Inspired by this natural mechanism, this paper explores commercial off-the-shelf (COTS) sensors capable of detecting similar tactile information when grasping objects, akin to the capabilities of human hands. The aim is to showcase the effectiveness of available COTS tactile sensor data, as individual sensor data and combinations in classification tasks. The results will demonstrate the potential of individual sensors as mechanoreceptor-inspired tactile receptors performing hardness classification using various machine learning algorithms. Furthermore, this study aims to investigate the potential enhancement in accuracy by combining data from each tactile sensor, demonstrating how collective features from each sensor perform in hardness classification. This is expected to deliver a competitive agile solution for hardness classification that can be incorporated and scaled in robotic systems particularly where vision base solutions are unsuitable.

This paper comprises various sections that identify the required tools, methods, and adaptations to execute machine learning on the collected data, described in the following manner. Section 2 provides the foundational understanding of mechanoreceptors, tactile sensors, and material classification, drawing inspiration for this paper. It serves as the basis for the selection process of commercial off-the-shelf (COTS) sensors, leading to their

selectivity based on the functionality of mechanoreceptors. Section 3 presents an approach for COTS sensors identification based on mechanoreceptors inspiration, a further stage to implement sensors with grippers and ML steps to implement an algorithm on data structure obtained from COTS. Section 4 presents the design of the experiment which show case data collection action which connects the approach and experiment setup in collecting data from selected sensors, forming different data configurations from COTS. It explains in depth the grasping method performed using pneumatic grippers utilising pressure to control the gripper system and further also explains how the object was selected and prepared using Shore's qualitative hardness scale (H,S,F,ES) and how data collection was performed. Section 5 showcases three results: 1st for binary (H,S) classification for different ML algorithms with different configurations of sensor data, 2nd for ternary (H,S,F) classification, 3rd for best algorithm outcome with quaternary (H,S,F,ES) classification.

2. Background

Human hands have exceptional tactile capabilities and serve as a significant inspiration for advancing robotic perception. Mechanoreceptors are special sensors/receptors in human skin that detect various tactile sensations like pressure, force, and vibration. They play a crucial role in our ability to perceive and interact with the world around us. Understanding how mechanoreceptors work and their capability to detect hardness is essential for robotics research. Replicating these sensors in robots can provide the capability to perceive and interpret tactile information while grasping. This would enable robots to better understand objects with more precision and perform tasks that require sensitivity to hardness.

2.1. Mechanoreceptors

How a human detects the property of an object using touch, grasping, and picking is based on receptors (biological transducers) which convert any touch to stimulus into tactile information to signal to the brain to judge the object's property. Mechanoreceptors are fundamental sensors that detect different physical/mechanical properties of objects (tactile information) mainly by four receptors [35,36]. Figure 1 illustrates the following: Four of them have the capabilities to detect different physical properties like force, vibration, deformation, indentation, etc. (1) Merkel Cells (Discs): Merkel cells respond to changes in pressure and texture, allowing them to detect variations in surface features such as roughness and indentation, informing of force/pressure. (2) Meissner's Corpuscles: Meissner's corpuscles are sensitive to light touch and low-frequency vibrations, which detect any changes in surface texture and shape. (3) Pacinian Corpuscles: The primary function is to detect sudden changes in vibration and pressure. (4) Ruffini Endings: Ruffini endings are sensitive to sustained pressure and skin stretch, enabling them to detect changes in skin deformation caused by object contact. Based on current understanding, mechanoreceptors comprise tactile receptors that receive multiple pieces of information while grasping or touching the object which processes the tactile information through neurons to the brain to understand the properties and define it or store it in the brain for further reference [37]. Mechanoreceptors are responsible for detecting mechanical stimuli, including pressure, vibration, and deformation, thereby providing humans with valuable tactile information [6,35–37]. In the context of tactile perception, mechanoreceptors play a crucial role in assessing the hardness and softness of objects by responding to various tactile signals. Four important aspects understood from receptors illustrated in Figure 1 are that touch, force, vibration, and sliding are the key physical properties that make humans understand if the object is hard or soft or flexible while grasping or squeezing.

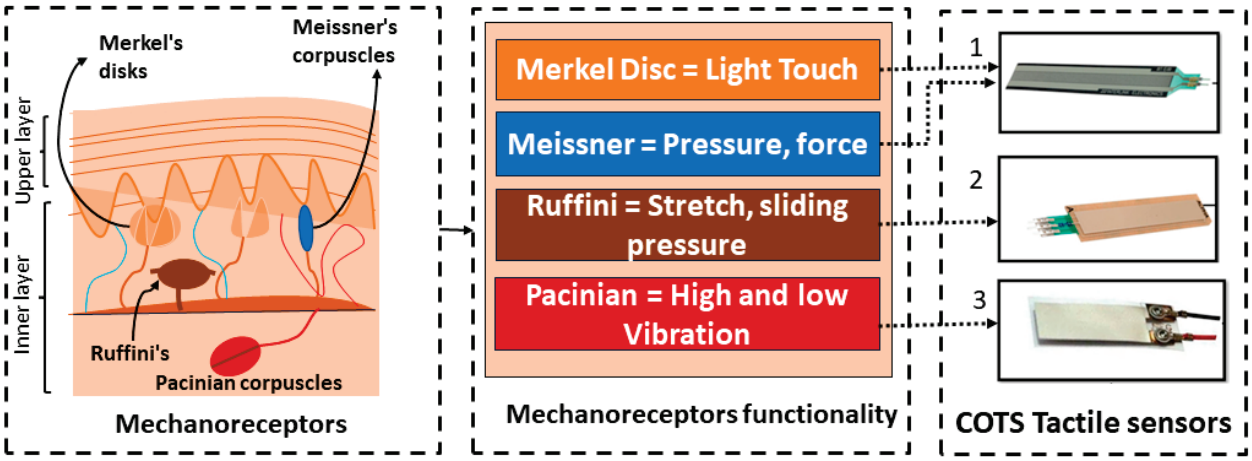


Figure 1. Showcases the mechanoreceptors' architecture and their functionality in detecting different tactile information, where COTS tactile sensors are mapped based on their functionality: (1) FSR force sensor, (2) potentiometer Softpot membrane, (3) piezoelectric thin-film vibration sensor.

2.2. Tactile Sensors

Material classification in robotics is one of the important aspects of development which helps robotic grippers in different environments to handle different objects or perform manipulation. This objective is only possible if sensors embedded within grippers have the capability to detect tactical information to recognize object properties. Material classification can be further subclassified into hardness classification and texture classification. Currently, texture classification has been explored mostly, but hardness classification has not been explored much with different tactile sensors and off-the-shelf sensors. Sensors used in literature are highly customized with shape and dimension which takes time and cost to develop. Parallel research is now more focused in developing sensors that have human capabilities in detecting object characters like texture, hardness, roughness, softness, slippery, etc. [1]. Engineering tools and methods have been developed which have shown some capability to show response type and functionality to human receptors [15–21] but it remains challenging to develop a system or sensors that have the same capability as humans. In the literature, various types of sensors are discussed with respect to their cost, which often varies depending on the technology and complexity involved. Optical sensors and bioinspired sensors are generally considered high-cost options due to their intricate designs and advanced functionalities. Microfluidic sensors and MEMS-based sensors fall into the moderate- to high-cost category, as they involve sophisticated fabrication processes and specialized materials. Similarly, grid capacitive, piezoelectric, and resistive tactile sensors are also moderately priced, reflecting the complexity of their construction and sensing mechanisms. Neuron memristors and ionic tactile sensors are relatively newer technologies, occupying the moderate cost range as they undergo further development and refinement [5,29,38,39]. However, all of them face limitations including high cost, complexity, sensitivity to environmental factors, integration challenges, and training issues with ML models due to the need for extensive datasets and iterative refinement filtration, etc. In comparison, commercially available off-the-shelf (COTS) sensors are easy to use, cost-effective, ready to deploy, flexible to adjust to most grippers, and suitable for exploring with different robotics grippers as tactile sensing techniques. Moreover, these sensors, such as vibration sensors, force-sensitive resistors (FSR), and soft potentiometers, can also play a role as artificial mechanoreceptors based on functionality.

2.3. Material Classification

Material classification involves sorting objects based on various features, much like how humans recognize things by feeling their texture and hardness when they touch or hold them. While texture classification, edge detection, and object classification have

received considerable attention, hardness classification has been relatively less explored, despite its significant importance. In the field of material science and engineering, machine learning (ML) techniques have proven invaluable in determining material properties using various methods and approaches, such as the Shore scale taxonomy [7,10,12,36,38,39]. ML enables systems to learn and recognize materials based on their hardness using techniques like grasping, indentation, and resistance methods. By using extensive datasets, ML models discern patterns and correlations, preparing them to accurately classify materials into predefined hardness classes. This iterative refinement process ensures their reliability, aiding industries in material selection and enhancing robotic grasping mechanisms through tactile sensing. Inspired by human grasping mechanisms, ML algorithms serve as analytical tools, processing tactile information to identify object properties. Robotics systems lack the analytical capabilities of the human brain to recognize object properties. However, through the integration of sensor data and machine learning (ML), this approach becomes feasible in advanced tactile sensing for robotics.

These algorithms are built upon various mathematical formulations, which serve as the backend processes. By encapsulating them as functions within ML libraries and implementing them through Python coding, they become readily deployable for data analysis. This facilitates advancements in material classification and robotics. The hardness classification process initiates with a long process of data collection through experiments, gathering information on material properties alongside corresponding hardness measurements. Thorough preprocessing follows, ensuring the quality and integrity of the dataset, while relevant features indicative of hardness properties are selected or extracted to facilitate accurate classification. Subsequently, an appropriate ML model is selected and trained using the pre-processed dataset, allowing it to learn patterns and relationships between input features and hardness classifications. Finally, the trained model's performance is evaluated using established metrics, and iterative refinement and optimization processes are employed to enhance its accuracy and effectiveness in hardness classification tasks. Different types of ML algorithm have been deployed for the purpose of classification which were presented as state-of-art in paper [8,9,24,30–32]. To see how multiple ML algorithms perform in terms of accuracy of hardness classification, grasping data from COTS sensors was adapted against different objects selected and prepared based on the Shore hardness scale.

2.4. Shore Hardness Scale

The Shore hardness scale is a measurement technique used to assess the hardness or softness of material on a wider scale. The Shore hardness scale was adapted in several cases of classification [10] to match the value of an object as a comparison to testify the assumption made with the object as soft or hard while testing. In many classification approaches, hardness and softness were considered because of easiness in testing. It provides a standardized method for quantifying the resistance of a material by imposing force which works similar for humans while considering the object without noticing the values. There are different options of the Shore hardness scale, such as Shore A, Shore D, and Shore OO, each tailored to specific material types and testing conditions. The Shore hardness scale can be explored or used in different ways where one could select the material or object based on the Shore scale that could precisely describe that material shows this property and can be matched to a qualitative scale as extra soft, soft, medium soft, medium hard, hard, and extra hard. In this process, the qualitative scale was adapted to extend the classification beyond binary classification and adapt multiple objects based on the scale beyond hard and soft which could be flexible (deformable but with force). This approach also makes object adaption faster and gives precise results based on ML algorithms' outcomes if the extension of classes improves the outcome or not; also, there is the option to test wider unknown/new test objects' data.

2.5. Summary

Human hands hold exceptional tactile detection capabilities, such as pressure, force, and vibration, due to mechanoreceptors in the skin. These mechanoreceptors are crucial for understanding and interacting with the environment. Inspired by this natural mechanism, which involves converting tactile stimuli into neural signals, this research aims to use tactile capabilities of humans in robotic systems. Unlike existing methods that use highly customized and expensive sensors, this research explores the use of commercial off-the-shelf (COTS) tactile sensors, which are cost-effective and easy to integrate. Hardness classification is particularly important because it enables robots to handle and manipulate objects with precision and can provide capabilities to perform material-type identification and sorting, preventing damage to delicate items and ensuring a secure hold on harder objects, the same as how humans assess material properties through touch or grasping. This capability can improve robot–human interaction such as service robots or robotic assistants; understanding material hardness enhances their ability to perform tasks that involve human-like manipulation skills. By using machine learning (ML) techniques, robots can be trained to recognize material hardness, improving their grasping and manipulation abilities. The Shore hardness scale is employed to provide a standardized measurement of material hardness, aiding in the development of more effective robotic tactile sensing systems.

3. Approach

3.1. Identification of Mechanoreceptor-Inspired Tactile Sensors

Different types of complex and customized tactile sensors were identified in literature which were used in different application and classification tasks. New proposed tactile sensors like Biomimetic tactile sensors, artificial mechanoreceptor tactile sensors, grid-type piezoelectric sensors, and ionics tactile sensors may be under a commercializing state and are not so quickly available [7,13,15,38,40–42]. Based on the current state of the art, it is understood that artificial mechanoreceptors and tactile sensors share a common developmental base, detecting similar physical properties from stimuli. These sensors may vary in their single- or multi-dimensional array configurations. Many studies have focused on emulating spike patterns using artificial mechanoreceptors, with some applied in hardness classification tasks [39,43–45]. Currently, many of these advanced sensors are not available in thin film for multiple tasking within robotics applications. These types of sensors are very sensitive and can quickly degrade by certain actions of grasping and cannot withstand longer periods for data collection with multiple objects. These are advanced, but not much application has been proposed within robotics grasping or in classification. Despite numerous descriptions of artificial mechanoreceptors and multilayer sensors in literature, their use cases remain limited, with commercial availability posing another challenge. Proposed sensors as solutions are customized based on application which does not have further space or scope of development or availability [22,29,38,46]. Thin-film sensors are easy to embed in most robotics grippers, and based on market research, some of the sensors were identified which have some standard dimensions which can be used directly within the robotics application. Therefore, the focus shifts to available tactile sensors in the market, classified based on functionality mirroring natural mechanoreceptors illustrated in Figure 1. These include FSR force-resistive sensors (also known as piezoresistive sensors), vibration sensors (akin to piezoelectric sensors), and soft potentiometers sensors (such as distributed force-resistive sensors). These thin-film sensors can be easily integrated into hard or soft grippers to detect physical changes in force or distributed pressure. For this purpose, the Schunk gripper was adapted due to its capability to accommodate these sensors within its internal structure.

FSR sensors as mechanoreceptor 1 and 2 functionality: FSR sensors, also known as force-sensitive resistors, are tactile sensors designed to detect changes in resistance when force is applied. This change in resistance leads to an increase in voltage, which is typically scaled within the range of zero to five volts. The sensitivity of FSR sensors allows them to detect various levels of force, ranging from light touch to continuous pressure and even

high-impact forces, measuring up to 50N. This versatility makes FSR sensors well suited to emulate the functionality of two mechanoreceptors located on the top layer of the skin, namely the Meissner corpuscles and Merkel discs [47]. These receptors are positioned close to the skin's surface, enabling them to accurately perceive touch, pressure, or any form of deformation [47]. When an object deforms due to pressure or manipulation, the mechanoreceptors sense the deformations and send signals in proportion to the degree of deformity. Based on deformation values obtained, soft objects are classified as "soft" and hard objects as "hard" or flexible. This ability of pressure and resistance can be measured by FSR sensors which were also chosen.

Soft Potentiometer sensors as Mechanoreceptor 3 functionality: Potentiometer sensors' capabilities as mechanoreceptors have not yet been fully explored in terms of hardness classification. These sensors are the same as FSR sensors and exhibit a linear change in resistance across their surface. However, they possess a unique capability to detect slip, stretch, or sliding movements based on changes in resistance. This detection capability closely resembles the functionality of mechanoreceptors [6,47] known as Ruffini endings. In this scenario, this off-shelf tactile sensor has the capability to detect changes that are similar in mechanoreceptors to analyse object characteristics or parameters in terms of hardness classification. While squeezing, object deformation will produce stretch across the sensor and will give different output volts as signatures for different objects which will help to classify objects based on hardness.

Vibration sensors as Mechanoreceptor 4 functionality: Vibration sensors, functioning the same as mechanoreceptors, play an essential role in neural pattern generation, classification, and the development of grid sensor architectures. Piezoelectric vibration sensors serve as essential tools for detecting impact and vibrations across a wide spectrum of frequencies. This enables the collection of data crucial for pattern generation and classification tasks. While these sensors may not replicate the intricate architecture of mechanoreceptors, they offer functionality similar to Pacinian receptors, producing data vital for classification purposes. Vibrations generated upon object impact are transmitted to mechanoreceptor junctions. As these mechanoreceptors [6,47] deform in response to detected vibrations, they send electrical impulses to the brain for processing. Higher frequency vibrations typically indicate harder materials, requiring more energy to induce, while lower frequencies correspond to softer materials. This same functionality and process can be performed using thin-film vibration sensors to perform hardness classification.

3.2. Hardness Classification Using Mechanoreceptor-Inspired COTS Sensors

Figure 2 illustrates the proposed approach and steps involved, which are essentially divided into four parts. Firstly, it involves the sensors that are bio inspired by mechanoreceptors. Choosing the right gripper based on dimensions, adapting the grasping method (mechanical resistance) to create an impact on the sensors while grasping objects the same as human pinch grasping, and object selection on the Shore hardness qualitative scale are illustrated in Figure 2. Secondly, it utilizes a machine learning approach to analyse data from sensors, including F-force data, V-vibration data, and P-potentiometer data. Initially, single-sensor (F), (V), and (P) data were considered for analysis, and then different configurations of sensor data were formed from each sensor. This also shows how off-shelf tactile sensors data, selected based on bio-inspired mechanoreceptors, perform a hardness classification. This understanding indicates how an individual and combination of sensors can yield accurate results when attempted. Additionally, it offers insight into how mechanoreceptor-inspired tactile sensors, both individually and in combination, can classify hardness effectively. Accuracy from COTS tactile sensors will set a benchmark for further exploration in layered sensor technology for future scope. These accuracy score data showcase how grasping-based tactile information achieved from sensors in volt as a value can be used to achieve a hardness classification to evaluate their performance. Decisions are made based on the accuracy of each configuration in multi-class scenarios, including dual combinations like (F,V), (F,P), and (P,V), as well as the three-sensor combina-

tion (F,V,P). While binary classification has been explored in the literature, this approach also involves different sensors' combination approaches to perform binary (2 classes object), ternary (3 classes object), and quaternary classification (4 classes object) based on object data obtained from testing and Shore hardness scales.

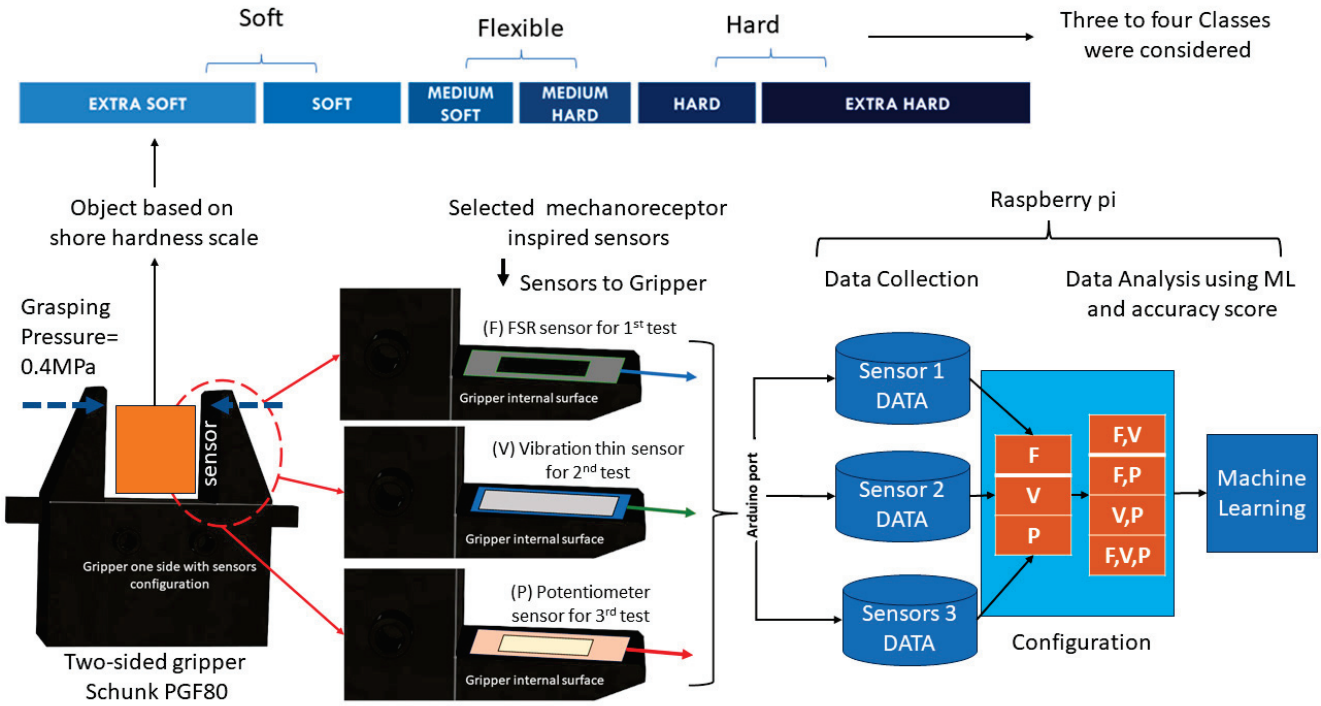


Figure 2. Research approach illustrating firstly object selectivity based on Shore scale considering three to four classes in object; secondly selected sensors embedded on gripper one side; thirdly sensors connected to Arduino to raspberry-pi for data collection; lastly from F,V,P data different set of configurations created to investigate accuracy score using multiple machine learning approaches.

3.3. Machine Learning Approach for Hardness Classification

Figure 3a illustrates the process followed in the machine learning approach and Figure 3b shows the data structure of configuration used to train the ML. Machine learning algorithms are essential for hardness classification tasks. These algorithms act like a brain analyser, analysing different types of data such as numeric, text, images, and more. In the context of hardness classification, it helps in understanding the features of datasets obtained from tactile sensors or any other source. The algorithms use various techniques, including supervised learning, where they are trained on labelled data to predict hardness classification. Examples of such techniques are decision tree classification, random forest classification, nearest neighbour classification, including support vector classification, and others. During training, the algorithms learn patterns and relationships within the data, allowing them to accurately classify new inputs. They analyse features extracted from the data to predict whether a material is hard or soft or based on scale. Previously bio signal, FFT, and other digital data were used in relation to objects to train ML [2,38]. In this approach, COTS sensors data were formed in different combinations to train the ML algorithms to understand which sensors' configuration out of F,V,P, (F,V), (F,P), (P,V), (F,P,V) trains ML well, and based on test data, which ML algorithm accuracy comes out to be the best. There are some steps that need to be taken before deploying the machine learning algorithm which are as follows:

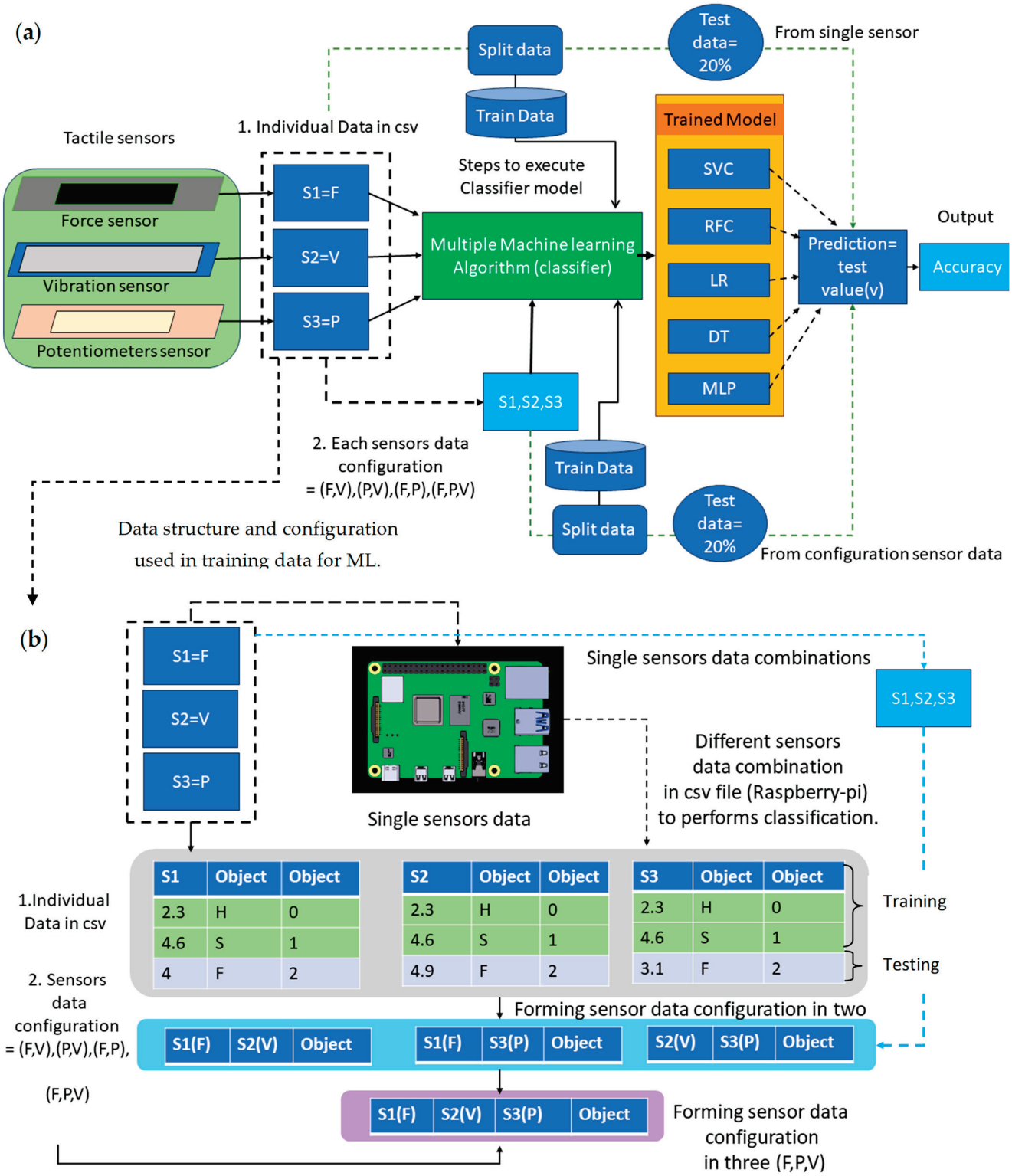


Figure 3. (**a**) Illustrates machine learning framework for COTS sensors data. Diverse machine learning algorithms are employed to train models using these data. The datasets utilized comprise (1) individual sensor data and (2) combinations of sensor data paired with object information denoted as (H,S) and (H,S,F). Subsequently, the test values are inputted into each algorithm, and the accuracy of their outputs is evaluated to test their efficacy. (**b**) Illustrates the data structure of various configurations saved within Raspberry Pi to train ML algorithms to investigate hardness classification.

Python Libraries: Machine learning analysis was conducted using Python using Google Collaboratory. To process all analysis, there were several Python and machine learning libraries which needed to be declared in beginning of the code. In general, libraries like Sklearn, NumPy, Pandas, and Matplotlib were declared in the first step [48–51].

Data importing: The initial step in analysis involves importing the dataset into the Python environment. This dataset serves as the foundation for machine learning tasks, containing the necessary information for training, and evaluating classifier models with a feature variable and a target variable. In this case, Figure 3b showcases different S1, S2, S3 data in single CSV file with the column of sensors' value in volts as X variable and target variable having the information of objects' type as H as 0, S as 1, F as 2 called as Y variable. These data were called and processed and were performed using the Python library called 'Pandas'. In a further case, two combinations (S1,S2), (S2,S3), (S3,S1) of dataset were combined with the help of 'Pandas' to further follow the combination approach.

Preprocessing data: Prior to training machine learning models, it is essential to preprocess the dataset to ensure compatibility with the chosen algorithms and improve model performance. One common preprocessing step involves encoding categorical variables into a numerical format, which is achieved using techniques such as label encoding, also illustrated in Figure 3b. In this case, the target variable needs to be mapped into numerical values using a label encoder, whose function is to transform string values like H as 0, S as 1, and F as 2, in case of ES (extra soft) as 3. Additionally, preprocessing may involve handling missing values, scaling features, and other data transformations required based on dataset preparation using the Imputer library.

Splitting data: To evaluate the performance of machine learning models accurately, it is crucial to divide the dataset into separate training and testing subsets. This process, known as data splitting, helps assess the generalization ability of the trained models by evaluating their performance on unseen data. In this case, data from the configuration F,V,P, (F,V), (F,P), (P,V), (F,P,V) was split accordingly to train the ML algorithm. Typically, a portion of the dataset is reserved for training, while the remaining portion is used for testing. In this case, 80% of the data were used for training and 20% of the data were used for testing, as illustrated in Figure 3a.

Importing ML algorithm: With the dataset prepared and split into appropriate subsets, it further proceeds to import the machine learning algorithms like SVC (support vector classifier), RFC (random forest classifier), DT (decision tree), LR (logistic regression), and MLP (multilayer perceptron) for data analysis. Depending on the nature of the problem and the characteristics of the dataset classifier, the algorithms found to be most suitable are used for training predictive models. As these algorithms take less computational time, they are easy to deploy and analyse in comparison to any form of deep learning or neural network.

Training ML: Once the algorithm(s) are imported, data split into X (sensors value) from different configurations F,V,P, (F,V), (F,P), (P,V), (F,P,V) and Y (target feature = object type (like H as 0, S as 1, and F as 2)) are illustrated in the Figure 3a,b variable used as training data to train the models, where the models learn patterns and relationships within the training data as sensors' values and number of object types. This process involves feeding the training data into the algorithm(s) and iteratively adjusting the model parameters to minimize a predefined loss function. Through this iterative optimization process, the models learn to make predictions or classify new data based on the patterns observed in the training set.

Trained model: After completing the training phase, trained machine learning models were obtained that capture the discerned patterns and correlations between tactile sensors' value and target variables (object types). These relationships form a crucial basis for distinguishing between (1) hard and soft materials within the dataset in case of binary classification and (2) three-object variables, hard, soft, and flexible, in case of ternary classification (H,S,F), as illustrated in Figure 3. With this information, the models are equipped to forecast outcomes for new, unseen instances, particularly in the context of hardness classification tasks.

Prediction: This prediction phase involves feeding unseen data into the trained models and obtaining output predictions based on the learned patterns. The predictions generated by the models represent how machine learning performed predictions based on dataset features and based on different models. In this case, two sets of data will be tested: (1) individual sensor data, and (2) combination of individual sensors from different ML algorithm outcomes in one file, as illustrated in Figure 3.

Validation using accuracy score: To evaluate the performance of the trained machine learning models, validation techniques such as accuracy score are illustrated in Figure 3a,b. In this case, data which was split will use 20% data for testing the model. In this way, models with new data will predict the outcomes as H or S object or for F object. This will be compared to the original test value which will give an accuracy score and performance by comparing it within literature. Also, the accuracy score provides a measure of the overall correctness of the model predictions. This is highlighted in the results section.

4. Design of Experiment

The design of the experiment illustrated in Figure 4 for conducting hardness classification involves both hardware and software components. Firstly, it was crucial to employ COTS tactile sensors to measure the impact of squeezing objects and collect data. This required performing grasping and resistance measurement actions to obtain values from three sensors inspired by mechanoreceptor tactile detection capability. Additionally, the gripper action was facilitated using air pressure, capable of withstanding a force of 0.4 MPa during grasp, controlled through Python coding via Raspberry Pi.

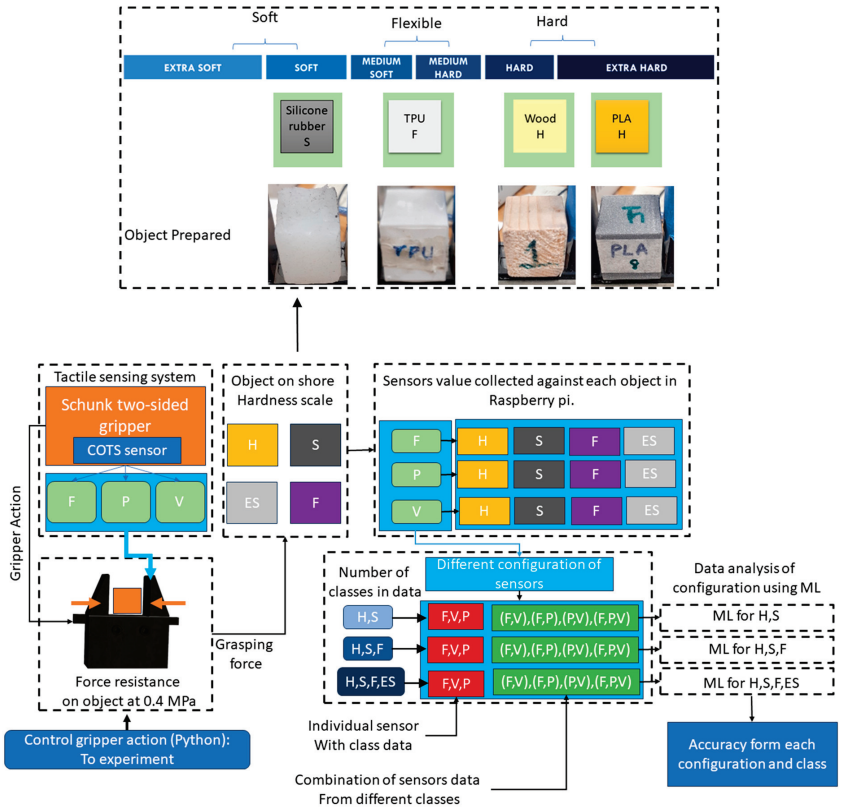


Figure 4. Design of experiment flow process for collecting data from COTS sensors and preparing different set of configuration data to machine learning analysis. Object preparation based on qualitative Shore hardness scale adopted in this study, highlighting the three to four classes considered: H (Hard), S (Soft), and F (Flexible), ES (extra soft). Objects representing each class are showcased based on the scale, with silicone rubber representing Soft, TPU representing Flexible, and Wood and PLA representing Hard materials.

The selection of an appropriate gripper was paramount, requiring compatibility with objects and sensors. A two-finger gripper was chosen for its ability to mimic human grasping and its economic viability for grasping various object types efficiently. The Schunk gripper, available in mechanical and pneumatic types, was selected for compatibility, with tactile sensors shaped to attach easily. Control of pneumatics involved the use of a pressure regulator, solenoid valve, and single electric valve. Integration of Arduino facilitated sensor connection and served as an ADC for data transmission. Additional hardware components, including a 3D printer for object creation and a screen with HDMI port for visualization, completed the experimental setup. Data collected from sensors, denoted as F (force), V (vibration), and P (potentiometer), were saved in Raspberry Pi. These data were further configured into various combinations, such as (F,V), (V,P), (F,P), and (F,P,V), to perform data analysis on individual sensor data and combinations thereof. To examine deeper into the obtained values from COTS sensors, a 'shore taxonomy' process was employed to analyse the deformation resistance to force based on the Shore hardness qualitative scale presented in Figure 4. Additionally, each configuration represented specific conditions for binary, ternary, and quaternary classification. For binary classification, the configurations included H (Hard) and S (Soft) classes. Ternary classification incorporated H (Hard), S (Soft), and F (Flexible) classes, while quaternary classification added an additional class, ES (Extra Soft). This approach was crucial for performing hardness classification against each object, providing insights into the behaviour of future applications in terms of sensor layer operation and the number of classes involved in classification.

4.1. Object Preparation

Illustrated in Figure 4: Objects representing different hardness classes (Soft, Flexible, Hard) are prepared according to the Shore hardness scale for testing. For silicone rubber, Eco flex 30 was used with mould of 3 $\times$ 3 cm size and filled with Eco flex liquid and left for around 3 h for best result. After extraction of silicone rubber, it was squeezed/pinch grasped to judge silicone rubber act as soft based on Shore scale. TPU (Thermoplastic polyurethane) is filament used in 3D printing to print objects. In this case, 3D object with 3 $\times$ 3 cm dimension was printed; based on property analysed, TPU was considered as flexible with medium hard and medium soft property. In case of wood, a piece was diced to form 3 $\times$ 3 cm wooden block. PLA (Polylactic Acid) is also a filament which was used to 3D print 4th object and analysed to be another hard object.

4.2. Hardware

Figure 5a illustrates complete collection of setups which showcase what tools and hardware were used. To conduct hardness classification by squeezing object, also known as grasping, and collecting data, it was necessary to have grippers that effectively have enough space to accommodate object and sensor on gripper. For this purpose, Schunk gripper as mechanical gripper/pneumatic gripper type was selected. Additionally, to collect data from tactile sensors, it was necessary for their shape to be compatible with sticking on the side of the pneumatic gripper. To control pneumatics, further tool requirement was to control air pressure, which was conducted using pressure regulator, solenoid valve, and single electric valve. Main air pressure supply was fed into pressure regulator using supply within IAC lab. To develop a system with both automatic and manual grasping capabilities, a device that could synchronize with Python and collect data simultaneously was required. This synchronization was achieved using Raspberry Pi, which served as the central hub with the resources to combine peripheral devices and control them through digital commands (high and low).

In addition to Raspberry Pi, an Arduino board was utilized to connect sensors and serve as an ADC (analogue to digital converter). This allowed the sensors to be connected to Arduino ports and subsequently linked to Raspberry Pi via serial communication. Other essential components included a pressure regulator to limit air pressure to a certain level, relay modules to control electric components, and solenoid valves to regulate air

pressure. An RS power supply was necessary to operate the electric valve and solenoid valve effectively. To perform gripping tasks, a Schunk pneumatic gripper was employed. Furthermore, a 3D printer was utilized to create testing objects from different materials, providing versatility in experimentation. Lastly, a screen with an HDMI port was utilized to visualize the outcome of the experiments, ensuring effective monitoring and analysis of the collected data.

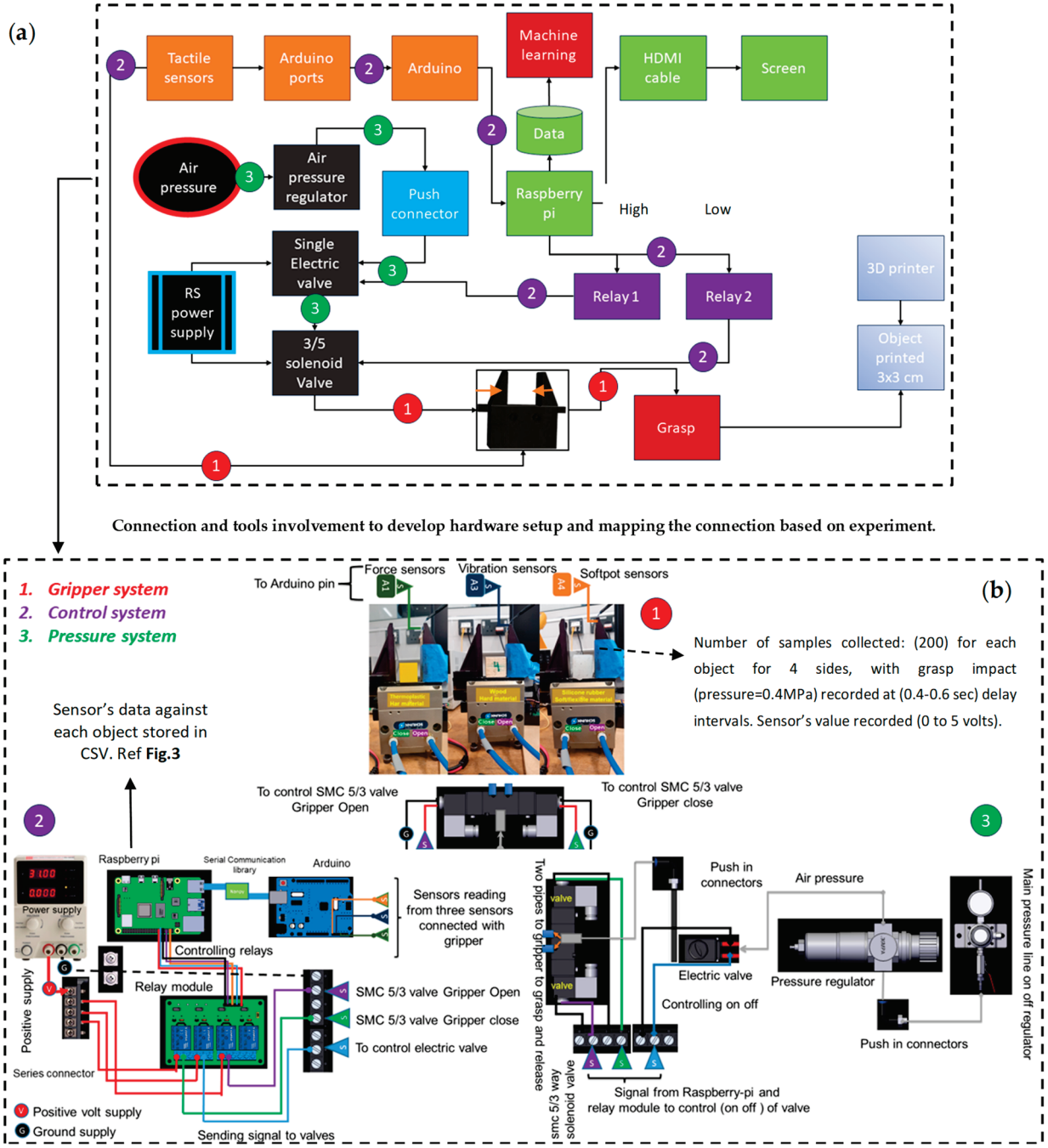


Figure 5. (**a**) Provides an overview or mapping of the experimental setup, demonstrating the connections among the tools involved in developing the prototype; (**b**) presents the complete experimental setup, showcasing three distinct systems: the Gripper system, Control system, and Pressure system. Additionally, three different tactile sensors are positioned on the side of the gripper to collect data alongside various objects. Data collected were saved in Raspberry Pi in CSV format. All data of sensors with objects have around 200 samples.

4.3. Software

To control grasping and collect sensor data for analysis using machine learning algorithms, various Python libraries and machine learning frameworks were employed. To establish communication between the Arduino and Raspberry Pi, the Nanpy library was used, allowing for integration of the two devices in a master–slave configuration to facilitate serial communication [52]. For data collection and processing, the Pandas library has capabilities to manage data and manipulate datasets. Additionally, for the analysis of collected data and the implementation of various machine learning algorithms, the Scikit-learn library was applied, offering a wide range of algorithm selectivity like SVM (support vector classifier), RFC (random forest classifier), and (DT) decision tree, and for measuring performance like accuracy and confusion matrix. Python 3, along with Python idle-Version 3.8 software, Jupyter Notebook Version 5, and Google Collaboratory, were employed for coding purposes. Each platform was selected based on its capabilities and suitability for specific tasks, ensuring efficient analysis and data collection across various scenarios.

4.4. Gripper System

In the exploration of various grippers documented in literature, diverse structures and use cases have been examined. However, the need arose for a general gripper that could mimic pinch grasping same as human hand operation, while also accommodating the dimensions of COTS sensors. Understanding the embedding issues of sensors when using different grippers and sensor variations was crucial. For the experiment depicted in Figure 5b, the Schunk PGF 80 gripper was utilized. One side of the gripper was equipped with sensors, with each sensor being swapped out during the experiment for each object. These sensors were connected to an Arduino to collect data, capturing tactile information from each stimulus object. A pneumatic system controlled by a control system regulated the opening and closing of the gripper by releasing air pressure through a pipe attached to the gripper. The selection of the gripper was based on the space available between its fingers, facilitating easy placement of objects. Additionally, the gripper's adjustable gripping range allowed it to grasp objects larger than its default size. Opting for a two-sided gripper enabled a pinch-type grasping similar to human finger dexterity, occupying minimal space, and swiftly integrating into any robotic arm. The selected sensors were easily installable in these grippers compared to custom-made ones, streamlining the experimental setup, and enhancing efficiency.

4.5. Control System

To control the gripper, an SMC solenoid valve which is controlled through relay module and with python code from Raspberry Pi sends high and low command to GPIO pins. This relay module controls the electric valve in pressure system which controls the main supply of air pressure system at decided delay of 0.4 to 0.6 s. Illustrated in Figure 5b, sensors connected in gripper also connected to Arduino which reads the analogue value from sensors and converts them into voltage on scale of 5 volts. Arduino is connected to Raspberry Pi which works as bridge to sensors and Raspberry Pi. With the help of Python library (Nanpy,) Raspberry Pi and Arduino is connected in master–slave configuration to perform serial communication. Sensors' data were saved in Raspberry Pi from Arduino as CSV file to use in material classification algorithm.

4.6. Pressure System

Air pressure was taken from IAC lab which was connected to central pressure system and pressure regulator with safety knob. This air pressure was passed through regulator which was set around 0.4 MPa approx. This passed through electric valve which was controlled by control system through relay module illustrated in Figure 5b. Main board developed in a way that it can control three multiple air pressure types within this module which can be useful for future scope in terms of pressure variation tests under gripper. In terms of open and close of air flow, described in Figure 5b within gripper system, it is

controlled through SMC 5/3 solenoid valve which when one side is open through relay module, gripper opens, and when other side valve is open, gripper closes or grasps. This was achieved by sending high(1) and low(0) command through Raspberry Pi through GPIO pins to 2nd and 3rd relay module.

4.7. Data Collection

The data collection process stated and described in Figure 5b demonstrates the comprehensive connectivity of each module, highlighting how sensors are linked to the gripper and subsequently connected to an Arduino for data transmission to a Raspberry Pi. These data are crucial for further analysis, particularly in material classification research. During the experiment, single sensors were sequentially deployed with the gripper to acquire data, as illustrated in Figure 5, showcasing the formatted data structure illustrated in Figure 4 for subsequent analysis. Each tactile sensor gathered data from three distinct classes of objects grasped by the gripper, with the Raspberry Pi Figure 4 storing these data for each object individually. Gripper grasps every object around 200 samples from each object and sensors with synchronized grasp and releases with duration around 0.4 s to 0.6 s. Utilizing Python's Pandas library, data collection was streamlined through commands that defined the object's state (H, S, or F) based on the Shore hardness quantitative scale. The collected data for each object, captured by three sensors individually connected to Arduino ports, was organized into separate data frames, and saved as CSV files on the Raspberry Pi. Additionally, string labels (H,S,F) were transformed into numerical values using label encoder to perform classifier analysis capabilities. The connection setup detailed in Figure 5a,b illustrates the integration of sensors with the gripper system during testing, providing clarity on how tactile sensor data were stored in CSV files on the Raspberry Pi. To analyse data, Figure 4 explains the data collection process and the resulting data structure and steps for ML algorithms analysis process illustrated in Figure 3.

5. Results Based on Machine Learning

5.1. Result on Hardness Classification Outcome Based on Two Classes (H,S)

In term of hardness-based classification when two classes were considered, (H,S) gives comparable accuracy, the same as in literature which used customized sensors. Figure 6 indicates that binary classification of individual sensors showcases accuracy ranging from 65% to 82% noticed from individual sensors (F), (P), and (V). Accuracy achieved from individual sensors represents that off-shelf tactile sensors can be used in hardness-based classification. In case of two classes (H,S), individual sensors can perform up to 80% of accuracy in predicting 20% of testing data as validation. Literature suggests 80 to 95% [2] of accuracy/prediction while using customized and complex tactile sensors. Ultrasonic sensors [32] used in literature describing the hardness and softness take too much space within the gripping area, which is difficult to adjust in different required environments, which is the same for customized sensors. In terms of the combination of sensors like (F,V) = 85%, (P,F) = 89%, and (V,P) = 87%, accuracy obtained shows that the combination of sensors increases the prediction accuracy which indicates that more tactile information as a feature increases machine learning accuracy. And out of multiple algorithm approaches, it shows that RFC performs best among others. In different configurations, RFC also performs better. Based on three sensors (F,P,V), the configuration achieves around 93% which highest among all configurations of sensors as features. This indicates that an increasing number of features (tactile information or value) from different sensors about objects increases the chance of obtaining optimum accuracy. This also showcases that COTS have the capability to perform hardness classification.

5.2. Hardness Classification Outcome Based on Three Classes (H,S,F)

Hardness classification using three classes showcases results obtained from multiple algorithms presented in Figure 7. In comparison to two classes (H,S), the outcome from three classes (H,S,F) has shown less accuracy among different configurations. In individual

sensors, accuracy drops below 70%; in the combination of two (F,V), (P,F), and (V,P), accuracy drops below 80%; only three combinations of individual sensor data (F,P,V) show 82% accuracy by the RFC algorithm which was best among all configurations. This outcome represents that on an increasing number of classes, accuracy decreases among ML algorithms. Accuracy obtained from three classes (H,S,F) was not investigated much in literature; this result shows that multiple sensors' data obtained from COTS as tactile information/ features can perform multiclass hardness classification with 80%+ accuracy with limited data.

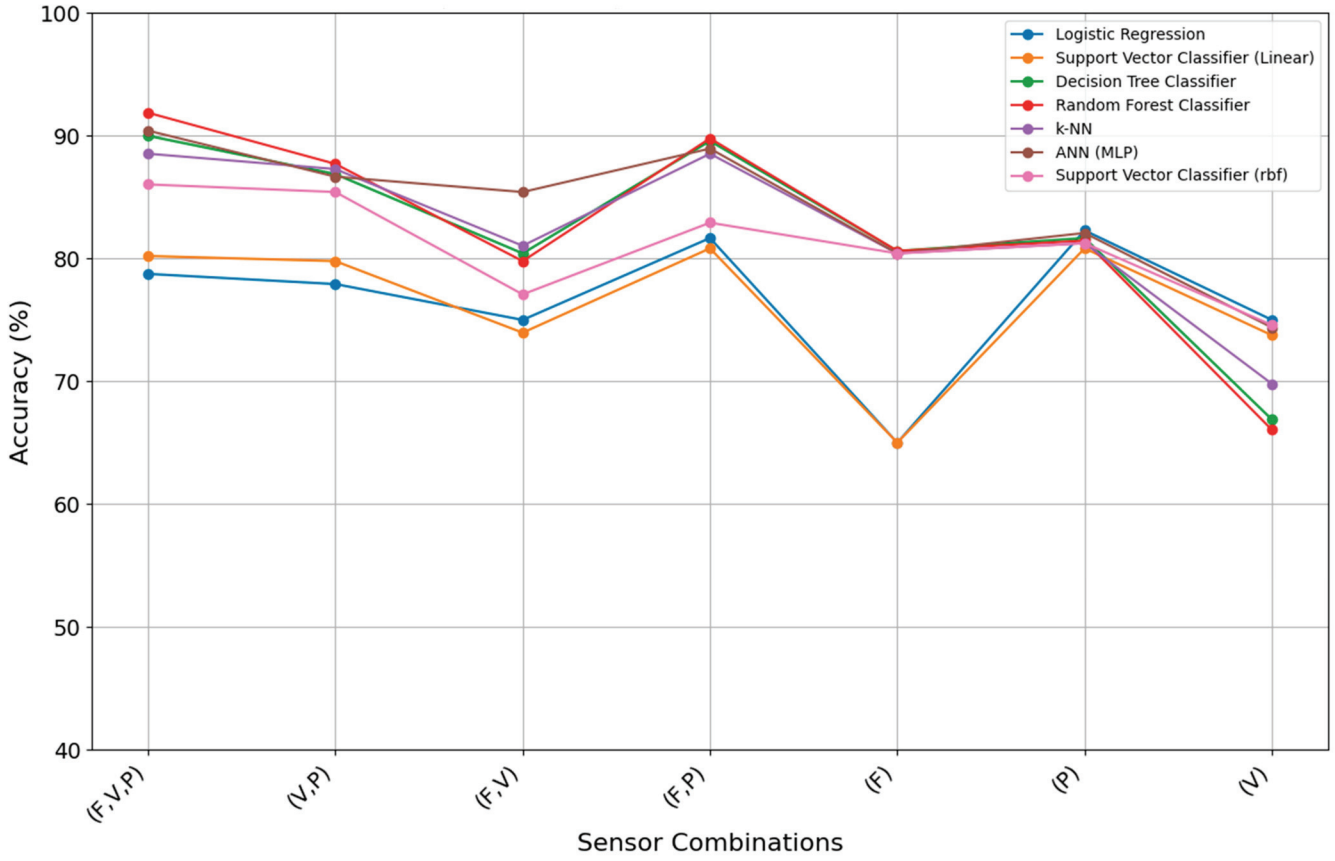


Figure 6. Describes the binary classification (H,S) outcomes among different sensor data configurations or combination using multiple algorithm outputs. Each algorithm is evaluated based on accuracy scores obtained from predictions on test data, which constitutes 20% of the overall dataset remaining unseen during training.

5.3. Result from Best Algorithm and Sensors Configuration—Multiclass Output

From class two (Section 5.1) and class three (Section 5.2) results, it been understood that the output of multiple algorithms indicates in most cases that RFC performs well. Multiple features from sensors with more tactile information (F,P,V) perform overall better. Three classes (H,S,F) with three sensors (F,V,P) configuration results outcome inspires a further look into including a fourth class from the Shore scale which is ES (extra soft). For ES objects, a white sponge was included, and ML models were trained again with all configurations. Figure 8 shows all results obtained from RFC and shows the case that binary classification remains optimum in most configurations where other classes' accuracy falls but shows the capability of performing hardness classification. In four object classes (H,S,ES,F), the (RFC) in (F,V,P) combination performs better among all other configurations achieving accuracy around 79%. Overall, this result also indicates that (F,V,P) combination performs better in all cases.

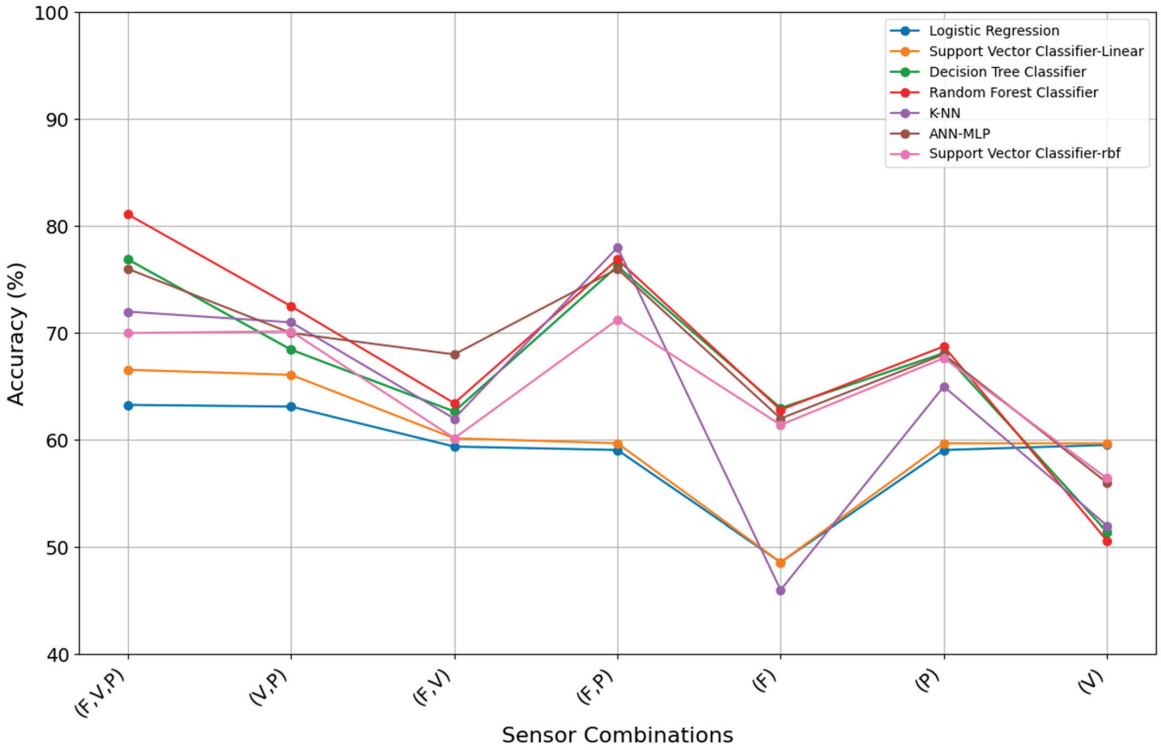


Figure 7. Illustrates the ternary classification (H,S,F) outcomes among different sensor data configurations or combination using multiple algorithm outputs. Each algorithm is evaluated based on accuracy scores obtained from predictions on test data, which constitutes 20% of the overall dataset remaining unseen during training.

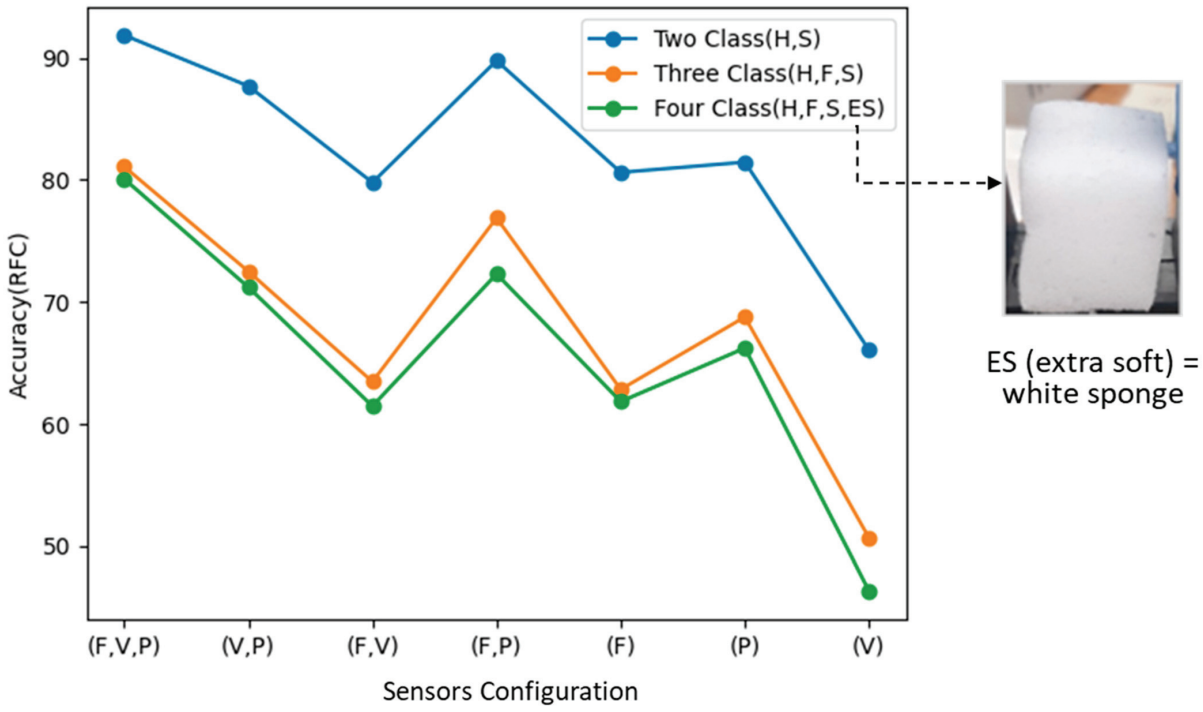


Figure 8. Illustrates the outcomes of multiclass hardness classification accuracy, demonstrating various sensor configurations. The best-performing configuration, utilizing three sensors (F,V,P), achieved optimal results, as highlighted by the outcomes obtained from the random forest classifier (RFC) algorithm.

6. Conclusions and Future Scope

This study demonstrates the feasibility and practicality of using commercial off-the-shelf (COTS) tactile sensors for multiclass hardness classification in robotic applications, specifically showcasing their scalability, cost-effectiveness, and integration ease compared to customized tactile sensors. It particularly underscores the advantages of configurations that incorporate three sensors (F,V,P), which consistently outperform simpler setups with accuracy rates between 80–92% across binary, ternary, and quaternary classifications. These findings highlight not only the feasibility of employing COTS sensors in varied robotic tasks but also their comparability in performance to more complex sensor systems documented in existing literature, where accuracy ranges from 50–97% (binary classification). Notably, the random forest classifier (RFC) was found to be particularly effective, likely due to its robust pattern recognition capabilities within the diverse data sets involving sensor values and target classifications.

In binary classification, individual sensors (F) and (P) have achieved accuracies above 80%, indicating that a single sensor may not be sufficient for comprehensive hardness classification. But individual sensors showcase the possibility of an accuracy score comparable to more complex grid or array sensors documented in the literature, which typically achieve accuracies between 50–90% [41]. This comparison highlights the viability of commercial off-the-shelf (COTS) sensors for rapid deployment in robotic applications, emphasizing their potential for broader use in texture classification and other areas. The selectivity of these sensors, inspired by human mechanoreceptors, is crucial. It enables the capture of diverse force, pressure, and vibration signals from various objects, enhancing hardness classification. This mechanoreceptor-based selectivity is especially beneficial when sensor data from multiple sources (F,V,P) are combined, suggesting future exploration into other bio-inspired sensors such as thermoreceptors for temperature, and optical receptors for light detection.

However, there are notable limitations to consider. The current testing procedures, which primarily involve uniformly shaped square objects, do not fully represent the range of real-world scenarios. The data collected from only four to five objects is limited in quantity, which may restrict the applicability of the tests in real-world and real-time scenarios. This limitation poses a challenge for performing hardness classification with COTS sensors, necessitating tests with a larger number of objects. Including more objects based on the Shore scale could either decrease or increase the accuracy of COTS sensors, which remains uncertain and highlights the need for future research. The resolution and sensitivity of the COTS sensors may not be sufficient for detecting fine differences in hardness, requiring tests with materials of various properties. Furthermore, testing conducted under controlled environments does not account for the variability in real-world conditions, such as temperature, humidity, and platform differences, which could affect sensor performance. Additionally, sensor alignment with various robotic grippers and different objects may yield varying outcomes. This underscores the potential for future work exploring the use of COTS sensors with dexterous robotic grippers to understand their performance across diverse applications. The use of single-sensor data, followed by their combination, can be time-consuming in real-time applications, posing a significant drawback. These factors emphasize the need for more comprehensive testing on objects of different sizes and shapes to accurately assess the capabilities and limitations of COTS sensors in various classification applications. Overall, this study has revealed several significant findings. Firstly, it demonstrated the feasibility and accessibility of hardness classification using commercial off-the-shelf (COTS) sensors, which require minimal processing time and are readily deployable in various robotics environments. Secondly, configurations using three sensors (F,P,V) consistently outperformed others, proving particularly effective in binary classification, although they were less effective in ternary and quaternary scenarios but performed optimally in comparison to others. Notably, among the various machine learning algorithms tested, the random forest classifier (RFC) exhibited optimal performance. This is likely due to RFC's ability to effectively discern patterns within the training data, especially

within the subset containing sensor values and target classifications. The achieved accuracy underscores the potential of COTS sensors, yielding results comparable to those documented in existing literature. These findings suggest significant potential for the extensive use of COTS sensors in robotic tactile sensing applications. Additionally, it has shown potential to explore layered or topology of COTS sensors to identify the optimal configurations using a bio-inspired (mechanoreceptor) approach. Future research will focus on analysing tactile information gathered collectively from layered sensors and performed in real-time predictions with unknown or new objects. In future work, we plan to expand the variety of materials testing to include broader parameters like textures, densities, gradient hardness, and composite structures to better evaluate the sensors' capabilities in diverse real-world scenarios. We may also explore advanced machine learning models, deep learning approaches, and ensemble techniques to improve the accuracy and robustness of multiclass classification tasks. In the case of improving ML models, various filtering techniques to enhance the quality of sensor data can also be considered in future work.

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

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

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

References

1. Jin, J.; Wang, S.; Zhang, Z.; Mei, D.; Wang, Y. Progress on flexible tactile sensors in robotic applications on objects properties recognition, manipulation and human-machine interactions. *Soft Sci.* **2023**, *3*, 8. [CrossRef]
2. Eguíluz, A.G.; Rañó, I.; Coleman, S.A.; McGinnity, T.M. Multimodal Material identification through recursive tactile sensing. *Robot. Auton. Syst.* **2018**, *106*, 130–139. [CrossRef]
3. Yi, Z.; Zhang, Y.; Peters, J. Bioinspired tactile sensor for surface roughness discrimination. *Sens. Actuators A Phys.* **2017**, *255*, 46–53. [CrossRef]
4. Li, G.; Liu, S.; Wang, L.; Zhu, R. Skin-inspired quadruple tactile sensors integrated on a robot hand enable object recognition. *Sci. Robot.* **2020**, *5*, 46–53. [CrossRef]
5. Li, F.; Wang, R.; Song, C.; Zhao, M.; Ren, H.; Wang, S.; Liang, K.; Li, D.; Ma, X.; Zhu, B.; et al. A Skin-Inspired Artificial Mechanoreceptor for Tactile Enhancement and Integration. *ACS Nano* **2021**, *15*, 16422–16431. [CrossRef] [PubMed]
6. Iheanacho, F.; Vellipuram, A.R. *Physiology, Mechanoreceptors*; StatPearls Publishing: Tampa, FL, USA, 2023.
7. Dahiya, R.; Oddo, C.; Mazzoni, A.; Jörntell, H. Biomimetic tactile sensing. In *Biomimetic Technologies*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 69–91. [CrossRef]
8. Amin, Y.; Gianoglio, C.; Valle, M. Embedded real-time objects' hardness classification for robotic grippers. *Future Gener. Comput. Syst.* **2023**, *148*, 211–224. [CrossRef]
9. Song, Y.; Lv, S.; Wang, F.; Li, M. Hardness-and-Type Recognition of Different Objects Based on a Novel Porous Graphene Flexible Tactile Sensor Array. *Micromachines* **2023**, *14*, 217. [CrossRef]
10. Qian, X.; Li, E.; Zhang, J.; Zhao, S.-N.; Wu, Q.-E.; Zhang, H.; Wang, W.; Wu, Y. Hardness Recognition of Robotic Forearm Based on Semi-supervised Generative Adversarial Networks. *Front. Neurorobot.* **2019**, *13*, 73. [CrossRef] [PubMed]
11. Jamali, N.; Sammut, C. Material classification by tactile sensing using surface textures. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; IEEE: New York, NY, USA, 2010; pp. 2336–2341.
12. Konstantinova, J.; Cotugno, G.; Stilli, A.; Noh, Y.; Althoefer, K. Object classification using hybrid fiber optical force/proximity sensor. In *2017 IEEE SENSORS*; IEEE: New York, NY, USA, 2017; pp. 1–3. [CrossRef]
13. Dai, K.; Wang, X.; Rojas, A.M.; Harber, E.; Tian, Y.; Paiva, N.; Gnehm, J.; Schindewolf, E.; Choset, H.; Webster-Wood, V.A.; et al. Design of a Biomimetic Tactile Sensor for Material Classification. *arXiv* **2022**, arXiv:2203.15941.
14. Madrigal, D.; Torres, G.; Ramos, F.; Vega, L. Cutaneous mechanoreceptor simulator. In Proceedings of the 2012 IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom), Kosice, Slovakia, 2–5 December 2012; IEEE: New York, NY, USA, 2012; pp. 781–786. [CrossRef]
15. Najarian, S.; Dargahi, J.; Mehrizi, A.A. *Artificial Tactile Sensing in Biomedical Engineering*; McGraw-Hill Education: New York, NY, USA, 2009.

16. Chun, S.; Kim, J.-S.; Yoo, Y.; Choi, Y.; Jung, S.J.; Jang, D.; Lee, G.; Song, K.-I.; Nam, K.S.; Youn, I.; et al. An artificial neural tactile sensing system. *Nat. Electron.* **2021**, *4*, 429–438. [CrossRef]
17. Pattnaik, D.; Sharma, Y.; Saveliev, S.; Borisov, P.; Akther, A.; Balanov, A.; Ferreira, P. Stress-induced Artificial neuron spiking in Diffusive memristors. *arXiv* **2023**, arXiv:2306.12853.
18. Kalita, H.; Krishnaprasad, A.; Choudhary, N.; Das, S.; Dev, D.; Ding, Y.; Tetard, L.; Chung, H.-S.; Jung, Y.; Roy, T. Artificial Neuron using Vertical MoS_2/Graphene Threshold Switching Memristors. *Sci. Rep.* **2019**, *9*, 53. [CrossRef] [PubMed]
19. Lucarotti, C.; Oddo, C.; Vitiello, N.; Carrozza, M. Synthetic and Bio-Artificial Tactile Sensing: A Review. *Sensors* **2013**, *13*, 1435–1466. [CrossRef]
20. Spigler, G.; Oddo, C.M.; Carrozza, M.C. Soft-neuromorphic artificial touch for applications in neuro-robotics. In Proceedings of the 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Rome, Italy, 24–27 June 2012; IEEE: New York, NY, USA, 2012; pp. 1913–1918. [CrossRef]
21. Bounakoff, C.; Hayward, V.; Genest, J.; Michaud, F.; Beauvais, J. Artificial fast-adapting mechanoreceptor based on carbon nanotube percolating network. *Sci. Rep.* **2022**, *12*, 2818. [CrossRef]
22. Kerr, E.; McGinnity, T.M.; Coleman, S. Material recognition using tactile sensing. *Expert Syst. Appl.* **2018**, *94*, 94–111. [CrossRef]
23. Cirillo, A.; Laudante, G.; Pirozzi, S. Tactile Sensor Data Interpretation for Estimation of Wire Features. *Electronics* **2021**, *10*, 1458. [CrossRef]
24. Drimus, A.; Kootstra, G.; Bilberg, A.; Kragic, D. Design of a flexible tactile sensor for classification of rigid and deformable objects. *Rob. Auton. Syst.* **2014**, *62*, 3–15. [CrossRef]
25. Gao, Z.; Ren, B.; Fang, Z.; Kang, H.; Han, J.; Li, J. Accurate recognition of object contour based on flexible piezoelectric and 9piezoresistive dual mode strain sensors. *Sens. Actuators A Phys.* **2021**, *332*, 113121. [CrossRef]
26. Suslak, T. There and Back again: A Stretch Receptor's Tale. 2015. Available online: https://era.ed.ac.uk/handle/1842/10474 (accessed on 1 April 2024).
27. Hosoda, K.; Tada, Y.; Asada, M. Anthropomorphic robotic soft fingertip with randomly distributed receptors. *Rob. Auton. Syst.* **2006**, *54*, 104–109. [CrossRef]
28. "Somatosensation-collective term for sensory signals from the body; Neuroscience Online. (n.d.). Somatosensory Processes (Section 2, Chapter 5). The University of Texas Medical School at Houston. Available online: https://nba.uth.tmc.edu/neuroscience/m/s2/chapter05.html (accessed on 17 June 2024).
29. Luo, S.; Bimbo, J.; Dahiya, R.; Liu, H. Robotic tactile perception of object properties: A review. *Mechatronics* **2017**, *48*, 54–67. [CrossRef]
30. Shimizu, T.; Shikida, M.; Sato, K.; Itoigawa, K. A New Type of Tactile Sensor Detecting Contact Force and Hardness of an Object. In Proceedings of the Technical Digest. MEMS 2002 IEEE International Conference, Fifteenth IEEE International Conference on Micro Electro Mechanical Systems (Cat. No.02CH37266), Las Vegas, NV, USA, 24–24 January 2002.
31. IEEE Robotics and Automation Society; Institute of Electrical and Electronics Engineers. *ICRA2017: IEEE International Conference on Robotics and Automation: Program, Singapore, 29 May–3 June 2017*; IEEE: New York, NY, USA, 2017.
32. Bouhamed, S.A.; Chakroun, M.; Kallel, I.K.; Derbel, H. Haralick feature selection for material rigidity recognition using ultrasound echo. In Proceedings of the 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 21–24 March 2018; IEEE: New York, NY, USA, 2018.
33. Schmidt, P.A.; Maël, E.; Würtz, R.P. A sensor for dynamic tactile information with applications in human–robot interaction and object exploration. *Rob. Auton. Syst.* **2006**, *54*, 1005–1014. [CrossRef]
34. Würschinger, H.; Mühlbauer, M.; Winter, M.; Engelbrecht, M.; Hanenkamp, N. Implementation and potentials of a machine vision system in a series production using deep learning and low-cost hardware. *Procedia CIRP* **2020**, *90*, 611–616. [CrossRef]
35. Dargahi, J.; Najarian, S. Human tactile perception as a standard for artificial tactile sensing—A review. *Int. J. Med. Robot. Comput. Assist. Surg.* **2004**, *1*, 23–35. [CrossRef] [PubMed]
36. Delmas, P.; Hao, J.; Rodat-Despoix, L. Molecular mechanisms of mechanotransduction in mammalian sensory neurons. *Nat. Rev. Neurosci.* **2011**, *12*, 139–153. [CrossRef] [PubMed]
37. Ding, S.; Bhushan, B. Tactile perception of skin and skin cream by friction induced vibrations. *J. Colloid. Interface Sci.* **2016**, *481*, 131–143. [CrossRef] [PubMed]
38. Yi, Z.; Zhang, Y.; Peters, J. Biomimetic tactile sensors and signal processing with spike trains: A review. *Sens. Actuators A Phys.* **2018**, *269*, 41–52. [CrossRef]
39. Kappassov, Z.; Corrales, J.-A.; Perdereau, V. Tactile sensing in dexterous robot hands—Review. *Rob. Auton. Syst.* **2015**, *74*, 195–220. [CrossRef]
40. Rahiminejad, E.; Parvizi-Fard, A.; Iskarous, M.M.; Thakor, N.V.; Amiri, M. A Biomimetic Circuit for Electronic Skin with Application in Hand Prosthesis. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2021**, *29*, 2333–2344. [CrossRef]
41. Weng, J.; Yu, Y.; Zhang, J.; Wang, D.; Lu, Z.; Wang, Z.; Liang, J.; Zhang, S.; Li, X.; Lu, Y.; et al. A Biomimetic Optical Skin for Multimodal Tactile Perception Based on Optical Microfiber Coupler Neuron. *J. Light. Technol.* **2023**, *41*, 1874–1883. [CrossRef]
42. Kerr, E.; McGinnity, T.M.; Coleman, S. Material classification based on thermal properties—A robot and human evaluation. In Proceedings of the 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 12–14 December 2013; IEEE: New York, NY, USA, 2013; pp. 1048–1053. [CrossRef]

43. Pestell, N.; Lepora, N.F. Artificial SA-I, RA-I and RA-II/vibrotactile afferents for tactile sensing of texture. *J. R. Soc. Interface* **2022**, *19*, 20210603. [CrossRef]
44. Liu, F.; Deswal, S.; Christou, A.; Sandamirskaya, Y.; Kaboli, M.; Dahiya, R. Neuro-inspired electronic skin for robots. *Sci. Robot.* **2022**, *7*, eabl7344. [CrossRef]
45. Luque, N.R.; Garrido, J.A.; Ralli, J.; Laredo, J.J.; Ros, E. From Sensors to Spikes: Evolving Receptive Fields to Enhance Sensorimotor Information in a Robot-Arm. *Int. J. Neural. Syst.* **2012**, *22*, 1250013. [CrossRef]
46. Li, P.; Ali, H.P.A.; Cheng, W.; Yang, J.; Tee, B.C.K. Bioinspired Prosthetic Interfaces. *Adv. Mater. Technol.* **2020**, *5*, 1900856. [CrossRef]
47. MacKinnon, C.D. Sensorimotor anatomy of gait, balance, and falls. *Handb. Clin. Neurol.* **2018**, *159*, 3–26. [CrossRef]
48. Hunter, J.D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
49. McKinney, W. Data Structures for Statistical Computing in Python. *SciPy* **2010**, *445*, 56–61. [CrossRef]
50. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
51. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn Res.* **2011**, *12*, 2825–2830.
52. Stagi, A. Nanpy Firmware. Available online: https://github.com/nanpy/nanpy-firmware (accessed on 21 May 2024).

electronics

MDPI

Article

Geometry Optimization of Stratospheric Pseudolite Network for Navigation Applications

Yi Qu [1,2,*], Sheng Wang [1,2], Hui Feng [1] and Qiang Liu [1]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; wangsheng@aircas.ac.cn (S.W.); fenghui@aircas.ac.cn (H.F.); liuqiang@aircas.ac.cn (Q.L.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: quyi@aircas.ac.cn

Abstract: A stratospheric pseudolite (SP) is a pseudolite installed on a stratospheric airship. A stratospheric pseudolite network (SPN) is composed of multiple SPs, which shows promising potential in navigation applications because of its station-keeping capability, long service duration, and flexible deployment. Most traditional research about SPN geometry optimization has centered on geometric dilution of precision (GDOP). However, previous research rarely dealt with the topic of how SPN geometry configuration not only affects its GDOP, but also affects its energy balance. To obtain an optimal integrated performance, this paper employs the proportion of energy consumption in energy production as an indicator to assess SPN energy status and designs a composite indicator including GDOP and energy status to assess SPN geometry performance. Then, this paper proposes an SPN geometry optimization algorithm based on gray wolf optimization. Furthermore, this paper implements a series of simulations with an SPN composed of six SPs in a specific service area. Simulations show that the proposed algorithm can obtain SPN geometry solutions with good GDOP and energy balance performance. Also, simulations show that in the supposed scenarios and the specific area, a higher SP altitude can improve both GDOP and energy balance, while a lower SP latitude can improve SPN energy status.

Keywords: stratospheric airship; pseudolite network; geometric dilution of precision (GDOP); energy balance; gray wolf optimization

1. Introduction

Stratospheric airships are flight vehicles that can keep their flight altitude by buoyance. They can reside in the lower portion of the stratosphere and perform station-keeping missions for a long time. This capability can provide a very efficient flight for many missions, such as science exploration, communications, earth observations, and navigations.

Researchers have proposed the concept of stratospheric pseudolites (SPs), which means to install transmitters on stratospheric airships to send out GPS-like signals to improve GNSS performance. Furthermore, a stratospheric pseudolite network (SPN) can be constructed by multiple SPs, which can provide independent positioning, GNSS augmentation and other services, especially in case of degraded visibility of GNSS signals. An SPN can provide many advantages such as wide coverage, long service duration, and flexible deployment; therefore, it has attracted abundant attention. Tsujii established a minimum configuration of a GPS/SP system to augment GPS and implemented experiments in both static and kinematic modes [1]. Dovis introduced an SPN system architecture, and discussed some key issues about SPN performance, including SP positioning, pseudo-ephemeris broadcasting, and GDOP improvement [2]. Zheng carried out simulations for SPN in urban areas and proved its effect on improving the horizontal dilution of precision (HDOP) and 3D positioning accuracy [3]. Chandu designed an SPN framework, described

its dataflow and mathematical model, and compared the feasibility of various SPN geometry configurations [4]. Dai presented SPN modeling strategies to deal with positioning error sources and proposed geometry optimization solutions for two application scenarios [5].

In previous research about SPN, geometry configuration has been identified as a critical factor affecting its positioning performance significantly. A series of approaches have been discussed to optimize the SPN geometry configuration, which can be divided into empirical methods and meta-heuristic optimization methods.

With empirical methods, Fateev suggested that an ideal pseudolite network should be composed of 5–10 pseudolites, which should be distributed along the edge of the service area and at different altitudes [6]. Sang studied the relationship between network geometry layout and geometric dilution of precision (GDOP) and illustrated three defective layouts that should be avoided in practice [7]. Hu, Gao, and Yang provided different geometry configurations for four SPs, five SPs, and six SPs, respectively, to minimize GDOP based on priori theories [8–10].

As to meta-heuristic optimization methods, Mosavi presented a pseudolite network geometry design approach with multiple evolutionary algorithms, including the genetic algorithm (GA), simulated annealing algorithm (SA), and particle swarm optimization algorithm (PSO) [11]. Shao offered a design strategy for pseudolite network geometry based on PSO and carried out indoor tests [12]. Tang put forward a multi-objective PSO algorithm for pseudolite network geometry design, whose purpose was to maximize the visual area while minimizing GDOP [13]. Yang adopted GA to search for optimal SPN to reduce the ephemeris error, ranging error, and positioning error in GNSS augmentation [14]. Chen utilized improved GA to select the best configuration aiming at enhancing accuracy in mobile positioning [15]. Song proposed an adaptive GA to realize geometry optimization under multiple constraints, especially under orographic terrain constraints and traffic facility constraints [16].

The studies listed above provided rich references for SPN geometry design. However, the energy balance requirement of SPN has rarely been analyzed, with even less consideration of the integrated optimization of SPN energy balance and GDOP.

In fact, the energy system of an SP is quite different from that of other aerial vehicles because it is always in the dynamic variation of energy production and energy consumption in its service duration. The balance between energy production and energy consumption is very vulnerable, since the energy gained from the solar arrays is quite limited, while the energy consumption due to resisting wind is enormous. Once its energy consumption exceeds energy production, an SP will encounter difficulties sustaining normal operation, making its service availability and continuity degrade greatly.

The main problem is that GDOP, SP energy production, and SP energy consumption are all governed by the SPN configuration. If GDOP is regarded as the sole objective during the course of SPN geometry design, an SPN unable to keep energy balance may be obtained. To avoid such an unpractical result, this paper proposes an SPN geometry design algorithm based on gray wolf optimization (GWO), pursuing the integrated optimization of GDOP and SPN energy balance.

This paper assumes the transmitting antennas onboard are directional antennas pointing down to the ground. Furthermore, this paper makes the following assumptions for an SP in its service duration, as Table 1 illustrates:

Table 1. Assumptions for an SP in its service duration.

Symbolic	Physical Meaning	Assumption
λ_j	longitude of the j-th SP	remain constant
Φ_j	latitude of the j-th SP	remain constant
h_j	altitude of the j-th SP	remain constant, 18–20 km as the preferred interval, and 20–28 km as the alternative interval [17,18]

Table 1. *Cont.*

Symbolic	Physical Meaning	Assumption
γ_j	yaw angle of the j-th SP	remain constant, do not affect the vector between a user receiver to the SP
α_j	pitch angle of the j-th SP	0
β_j	roll angle of the j-th SP	0
m_j	mass of the j-th SP	remain constant
J_{kaj}	energy spent on keeping flight altitude of the j-th SP	0
η_{vj}	photoelectric conversion efficiency	remain constant
J_{loss}	energy loss in the process of transfer, charging, storage, and discharging	0

The rest of this paper is organized as follows. The GDOP of SPN is analyzed in Section 2, an energy balance model of SPN is established in Section 3, an SPN geometry design algorithm based on GWO is proposed in Section 4, simulations and discussions are presented in Section 5, and conclusions and future works finally complete this paper.

2. GDOP of an SPN

GDOP is a widely used indicator in SPN performance assessment. It is defined as the statistics ratio of positioning accuracy, timing accuracy, and ranging accuracy. Given the same pseudo-range error, the smaller the GDOP is, the smaller the positioning error and timing error are.

In this paper, the GDOP of an SPN is defined as the average GDOP of multiple users in the SPN service area, which can be described by Equation (1).

$$GDOP_N = \frac{1}{n_u}\sum_{i=1}^{n_u} GDOP_i \tag{1}$$

In Equation (1), n_u represents the number of observers distributed in the service area. $GDOP_i$ represents the GDOP of the i-th user, which can be calculated by Equations (2)–(4) [19–21].

$$GDOP_i = \sqrt{tr(\boldsymbol{H}_i^T \boldsymbol{H}_i)^{-1}} \tag{2}$$

$$\boldsymbol{H}_i = \begin{bmatrix} a_{xi1} & a_{yi1} & a_{zi1} & 1 \\ a_{xi2} & a_{yi2} & a_{zi2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{xin_p} & a_{yin_p} & a_{zin_p} & 1 \end{bmatrix} \tag{3}$$

$$\begin{aligned} a_{xij} &= \frac{x_j - x_{ui}}{\sqrt{(x_j - x_{ui})^2 + (y_j - y_{ui})^2 + (z_j - z_{ui})^2}} \\ a_{yij} &= \frac{y_j - y_{ui}}{\sqrt{(x_j - x_{ui})^2 + (y_j - y_{ui})^2 + (z_j - z_{ui})^2}} \\ a_{zij} &= \frac{z_j - z_{ui}}{\sqrt{(x_j - x_{ui})^2 + (y_j - y_{ui})^2 + (z_j - z_{ui})^2}} \end{aligned} \tag{4}$$

In Equations (2)–(4), $\boldsymbol{H_i}$ represents the observation matrix of the i-th user, n_p represents the number of SPs in the network, a_{xij}, a_{yij}, a_{zij} represent the vector components between the i-th user receiver and the j-th SP, (x_{ui}, y_{ui}, z_{ui}) represent the position of the i-th user in the ECEF (Earth-Centered, Earth-Fixed) coordinate system, and (x_j, y_j, z_j) represent the position of the j-th SP in the ECEF coordinate system, which is gained from (λ_j, Φ_j, h_j) by coordinate transformation.

3. Energy Balance of an SPN

3.1. Energy Consumption of an SP

SP energy is used to support its equipment, such as the propeller, flight controller, TT&C (Telemetry, Tracking, and Command) system, and mission payloads. The propeller consumes a great deal of energy, and it is deeply affected by SP position and attitude. The energy consumption of the other equipment is slightly affected by SP position and attitude, and their requested power is assumed to be constant in this paper.

3.1.1. Energy Consumption of a Propeller

For the j-th SP, the power required by its propeller can be estimated by Equation (5) [22].

$$P_{ej} = \boldsymbol{T}_j \boldsymbol{U}_j / \eta_{pj} / \eta_{ej} \tag{5}$$

In Equation (5):

P_{ej} represents propeller power;

$\boldsymbol{T}_j$ represents thrust generated by the propeller;

$\boldsymbol{U}_j$ represents airspeed;

η_{pj} and η_{ej} represent propeller efficiency and motor efficiency, respectively, which can be assumed as constants according to the analysis in [22].

So, the energy consumption of the propulsion propeller of the j-th SP, represented by J_{ej}, can be expressed by Equation (6).

$$J_{ej} = \int_0^{t_s} \boldsymbol{T}_j \boldsymbol{U}_j / \eta_{pj} / \eta_{ej} dt \tag{6}$$

In Equation (6), t_s represents the station-keeping time of the SP.

To facilitate analysis, SP motion can be decomposed into motion along the axis direction and motion along the normal direction. The propeller is assumed to be able to generate thrusts along the axis direction and along the normal direction independently, as illustrated in Figure 1.

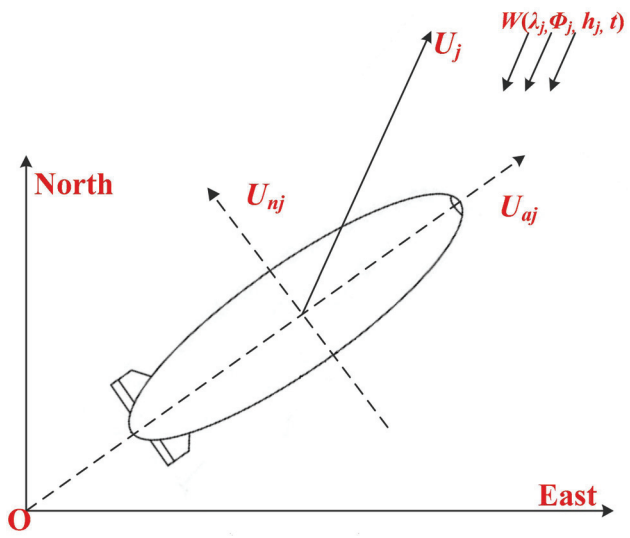


Figure 1. Illustration of an SP motion decomposition in the horizontal plane (since both pitch angle and roll angle of SPs are assumed as 0, this figure only illustrates the motion decomposition in the horizontal plane).

Then, Equation (6) can be rewritten into Equation (7) [23].

$$J_{ej} = \int_0^{t_s} (\boldsymbol{T_{aj}}\boldsymbol{U_{aj}} + \boldsymbol{T_{nj}}\boldsymbol{U_{nj}})/\eta_{pj}/\eta_{ej}dt \tag{7}$$

In Equation (7):
$\boldsymbol{T}_{aj}$ represents the axial component of thrust $\boldsymbol{T}_j$;
$\boldsymbol{U}_{aj}$ represents the axial component of airspeed $\boldsymbol{U}_j$;
$\boldsymbol{T}_{nj}$ represents the normal component of thrust $\boldsymbol{T}_j$;
$\boldsymbol{U}_{nj}$ represents the normal component of airspeed $\boldsymbol{U}_j$;
$\boldsymbol{T}_{aj}$ and $\boldsymbol{T}_{nj}$ can be estimated by aerodynamic resistance, and $\boldsymbol{U}_{aj}$ and $\boldsymbol{U}_{nj}$ can be estimated by local wind speed, as described in Equation (8).

$$\begin{cases} \boldsymbol{T_{aj}} = -\boldsymbol{D_{aj}} = -\frac{1}{2}\rho_{mj}V_j^{2/3}\boldsymbol{U_{aj}^2}C_{Dj} \\ \boldsymbol{U_{aj}} = -\boldsymbol{W}(\lambda_j, \phi_j, h_j, t)\cos(\gamma_j - \gamma_w) \\ \boldsymbol{T_{nj}} = -\boldsymbol{D_{nj}} = -\frac{1}{2}\rho_{mj}V_j^{2/3}\boldsymbol{U_{nj}^2}C_{Dj} \\ \boldsymbol{U_{nj}} = -\boldsymbol{W}(\lambda_j, \phi_j, h_j, t)\sin(\gamma_j - \gamma_w) \end{cases} \tag{8}$$

In Equation (8):
$\boldsymbol{D}_{aj}$ represents the aerodynamic resistance along the axis direction;
$\boldsymbol{D}_{nj}$ represents the aerodynamic resistance along the normal direction;
$\boldsymbol{W}(\lambda_j, \Phi_j, h_j, t)$ represents the wind speed at position (λ_j, Φ_j, h_j) and at time t;
ρ_{mj} represents the atmosphere density at altitude h_j;
V_j represents the volume of the j-th SP, and $V_j^{2/3}$ is used to estimate its reference area;
C_{Dj} represents its aerodynamic resistance coefficient;
γ_w represents the local wind direction angle.
Wind speed $\boldsymbol{W}(\lambda_j, \Phi_j, h_j, t)$, wind direction angle γ_w, and atmosphere density ρ_{mj} change enormously with position, as illustrated in Figures 2–4.

Figure 2 illustrates the meridional and zonal wind speeds in a specific area at an altitude of 20 km. It can be seen from Figure 2 that both meridional wind speed and zonal wind speed vary greatly with latitude in the area. The minimum zonal wind speed is about 8 m/s, while the maximum zonal wind speed has doubled to about 16 m/s. The meridional wind speed has undergone a directional reverse.

Figure 3 illustrates the meridional and zonal wind speeds at a certain location within the altitude interval of 18–30 km at four typical times in March, June, September, and December. From Figure 3, it can be seen that the zonal wind speed varies greatly with altitude changes, whose difference can reach tens of meters. Also, the meridional wind speed shows slight changes.

Figure 4 illustrates the atmospheric density variation of the U.S. standard atmosphere model at an altitude interval of 18–30 km. At an altitude of 18 km, the atmosphere density is about 0.12 kg/m^3, while at an altitude of 30 km, the atmosphere density rapidly decays to less than 0.02 kg/m^3. The attenuation of the atmosphere density exceeds 80%, reflecting a significant change.

These differences illustrated in Figures 2–4 can lead to a huge difference in the energy consumption of a propeller.

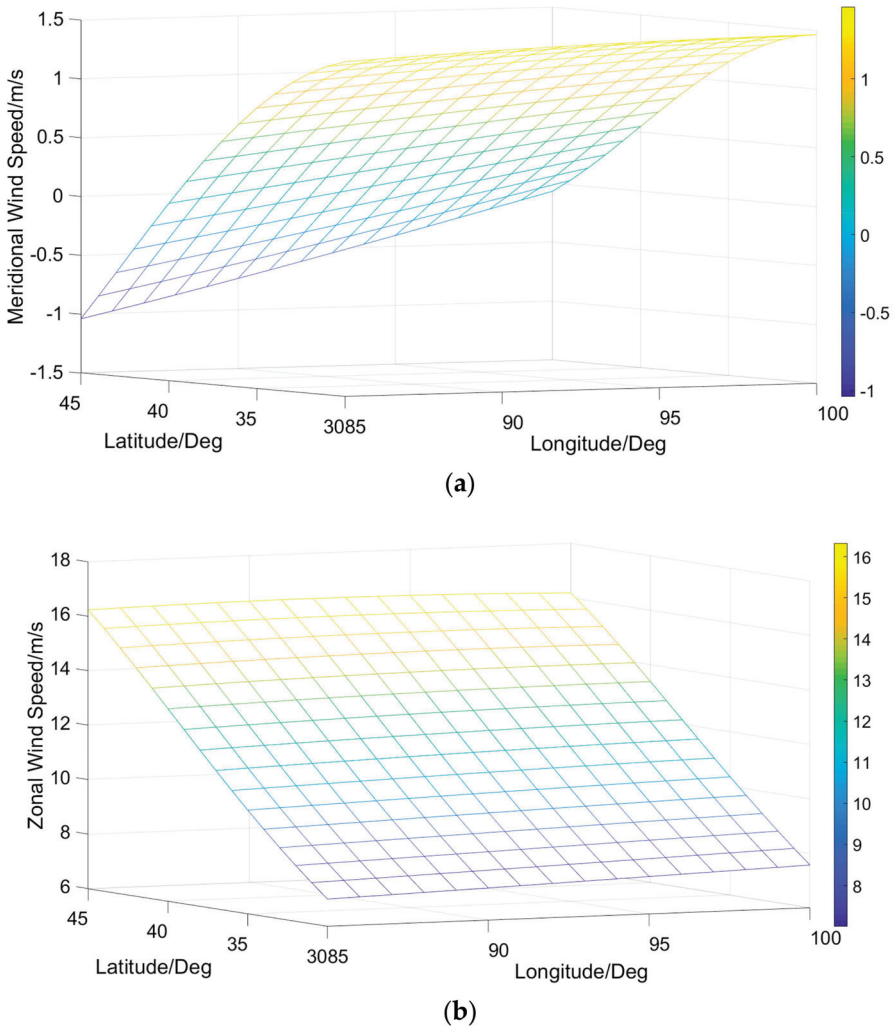


Figure 2. Horizontal wind speeds for a specific region at a certain time: (**a**) meridional wind; (**b**) zonal wind.

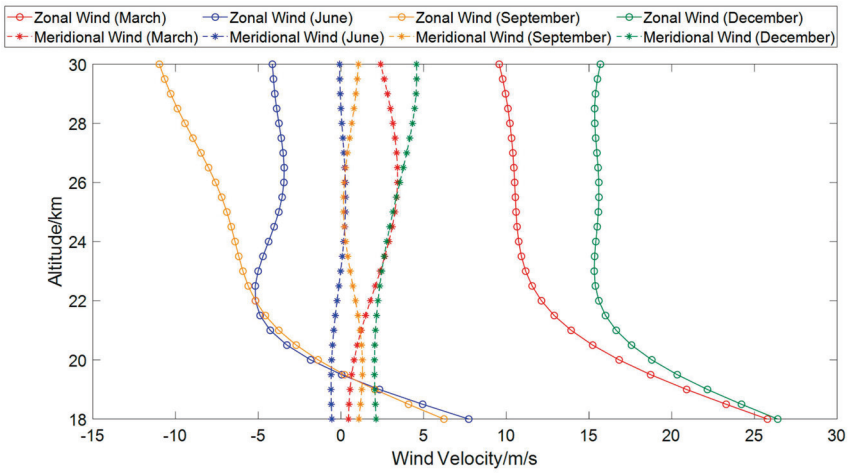


Figure 3. Horizontal winds at different altitudes for a specific region.

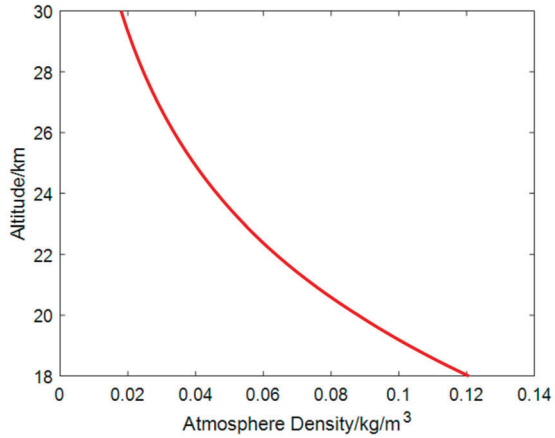


Figure 4. Atmosphere density from 18 km to 30 km according to U.S. Standard Atmosphere, 1976 [24].

3.1.2. Total Energy Consumption of an SP

The energy consumption of other equipment is only slightly affected by the SP position and attitude. Thus, their energy consumption is assumed to be a constant represented by P_{pj}. Their energy consumption, represented by J_{pj}, can be expressed simply by Equation (9).

$$J_{pj} = P_{pj}\, t_s \tag{9}$$

The total energy consumption of the j-th SP, represented by J_j, can be seen as the sum of propeller energy consumption and other equipment energy consumption, which can be expressed by Equation (10).

$$J_j = J_{ej} + J_{pj} \tag{10}$$

3.2. Energy Production of an SP

SP energy production relies on the solar arrays laying on its airship surface, which can convert solar radiation into electricity. Solar radiation harvested by the solar arrays can be divided into direct radiation, scattered radiation, and reflected radiation. Since the effect of scattered radiation and reflected radiation is far less than direct radiation, this paper emphasizes direct radiation and ignores scattered radiation and reflected radiation.

3.2.1. Solar Direct Radiation

The solar direct radiation intensity on the top of the atmosphere in the normal direction, represented by I_{top}, can be expressed by Equation (11) [25].

$$I_{top} = I_{SC}\, E_c \tag{11}$$

In Equation (11), I_{SC} represents the solar constant, and E_c represents a sun–earth distance correction, which can be expressed by Equation (12) [25,26].

$$E_c = 1 + 0.033\cos(2\pi d_n / 365) \tag{12}$$

In Equation (12), d_n represents the day number in a year.

The solar direct radiation intensity at different altitudes is affected by the atmosphere transmissivity, which can be expressed by Equation (13) [26].

$$I_{Dj} = I_{top}\tau_j \tag{13}$$

In Equation (13), I_{Dj} and τ_j represent solar direct radiation intensity and atmosphere transmissivity at altitude h_j, respectively. τ_j can be calculated by Equation (14) [27].

$$\tau_j = 0.56(e^{-0.65\lambda mj} + e^{-0.95\lambda mj}) \tag{14}$$

In Equation (14), λ_{mj} can be estimated by Equation (15) [27].

$$\lambda_{mj} = \frac{p_j}{p_0}[\sqrt{1229 + (614\sin\theta_{ej})^2} - 614\sin\theta_{ej}] \tag{15}$$

In Equation (15), p_j represents the pressure of the atmosphere at altitude h_j; p_0 represents the pressure of the atmosphere at sea level.

θ_{ej} represents the solar elevation angle at latitude Φ_j, which can be expressed by Equation (16) [26].

$$\sin\theta_{ej} = \sin\phi_j \sin\delta + \cos\phi_j \cos\delta \cos\omega_j \tag{16}$$

In Equation (16), δ represents sun declination; ω_j represents sun hour angle.

The sun declination δ is the angle of the sun above or below the equator plane. It changes with the date. It will reach its maximum value (23.45 degree) at the summer solstice in the northern hemisphere, and will reach its minimum value (-23.45 degree) at the winter solstice. δ can be roughly calculated by Equation (17) [25].

$$\delta = \begin{cases} 23.45\sin(2\pi(d_n - 81)/365) & d_n > 81 \\ 23.45\sin(2\pi(d_n + 284)/365) & d_n \leq 81 \end{cases} \tag{17}$$

The sun hour angle ω_j is the angle between the sun and the local meridian, which changes 15 degrees per hour and can be calculated by [25].

$$\omega_j = 15(t - 12) \tag{18}$$

3.2.2. Energy Production of a Solar Array

The solar array mounted on the airship surface can be divided into multiple cells to analyze its energy production precisely. If the area of cell k is represented by A_{jk}, the angle of the solar direction vector and the normal vector of cell k is represented by ψ_{jk}, and the output power of cell k produced by solar direct radiation, represented by P_{Djk}, can be calculated by Equation (19) [28,29].

$$P_{Djk} = I_{Dj}\, A_{jk} cos\psi_{jk} \tag{19}$$

The aggregate output power of all the cells in the solar array, represented by P_{PVj}, can be calculated by Equation (20).

$$P_{PVj} = \sum_{k=1}^{n_c} P_{Djk} \tag{20}$$

In Equation (20), n_c represents the number of solar array cells. The energy production of the j-th SP in its station-keeping time, represented by Q_j, can be expressed as Equation (21).

$$Q_j = \int_0^{t_s} P_{PVj}\eta_{Vj} dt \tag{21}$$

In Equation (21), η_{Vj} represents the photoelectric conversion efficiency of the solar array.

Figure 5 illustrates diurnal energy productions of an SP at different positions, reflecting energy production gaps caused by SP positions.

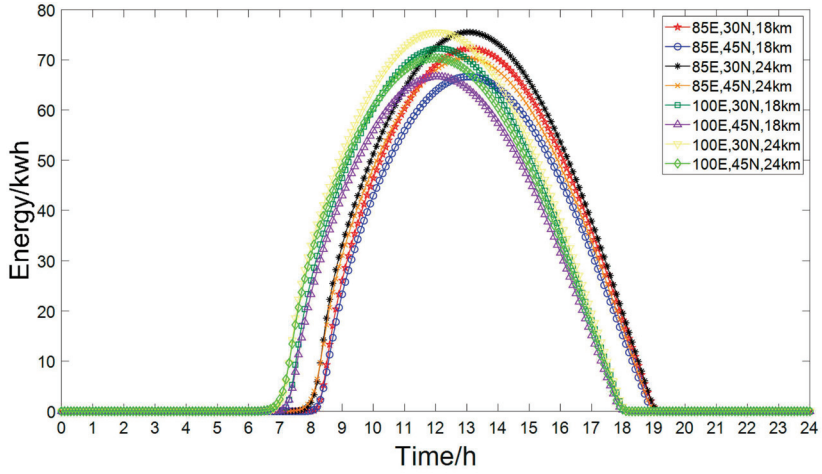


Figure 5. Comparison of energy productions of an SP at different positions on a specific day.

3.3. Energy Balance Indicator of an SPN

According to the analysis above, this paper proposes an energy balance indicator to assess an SP's energy status, which can be formulized as Equation (22).

$$B_j = J_j / Q_j \tag{22}$$

In Equation (22):

J_j represents the energy consumption of the j-th SP, which is defined in Equation (10);

Q_j represents the energy production of the j-th SP, which is defined in Equation (21);

B_j represents the ratio of energy consumption and energy production, reflecting the energy status of the j-th SP.

For an SPN, its energy balance can be assessed by the sum of all the individual energy balance indicators, which can be expressed as Equation (23).

$$B = \sum_{j=1}^{n_p} B_j \tag{23}$$

4. SPN Geometry Optimization Algorithm

4.1. Objective Function and Constraints

SPN design is a multi-objective optimization problem with multiple constraints. This study has two objectives: the first is to minimize SPN GDOP in a given area, which is defined in Equation (1); the second is to minimize the SPN energy balance indicator, which is defined in Equation (23).

Thus, the overall optimization objective function, represented by F, can be expressed as

$$\begin{gathered} F = w_1 GDOP_N + w_2 B \\ w_1 + w_2 = 1 \end{gathered} \tag{24}$$

In Equation (24), w_1 and w_2 represent the weight of GDOP and energy balance in the overall objective function, respectively. Their values can be adjusted within the interval [0, 1] according to requirements, keeping their sum as 1. If w_1 is set as 0, it means that GDOP will be ignored in the process of SPN geometry optimization, and if w_2 is set as 0, it means that energy balance will be ignored.

Two optimization constraints are emphasized in this study. The first is no co-location constraint, meaning that all the SPs in the network should not be deployed in the same position. It can be attributed to the large volume of airships, whose length can reach hundreds of meters. So, a distance is required between SPs to ensure safety, and the distance can be determined according to practical factors such as SP length. The second is

the individual energy balance constraint, requiring each SP in the network to maintain its individual energy balance, which means that for any $j \in [1, n_p]$,

$$B_j < 1 \tag{25}$$

4.2. SPN Geometry Optimization Based on GWO

GWO is a widely used optimization algorithm in many fields [30,31]. It is a meta-heuristic optimization algorithm developed by Mirjalili in 2014 that mimics the hunting behavior and leadership hierarchy of gray wolves [30]. Compared with traditional empirical methods, GWO requires neither gradient information nor continuous derivative of objective functions. Compared with other meta-heuristic optimization algorithms, such as GA, PSO, and SA, GWO has fairly competitive performances [30]. Therefore, GWO is employed in this paper to implement SPN geometry optimization.

In GWO, a solution for a problem is regarded as a gray wolf, and all the available solutions are regarded as a wolf population. Gray wolves in the population are divided into four types: alpha, beta, delta, and omega, representing the current best, sub-optimal, the third best, and other solutions, respectively. The search for the optimal solution is performed by three important strategies: approaching, surrounding, and attacking prey. For a detailed discussion about the strategies, please refer to [30].

In this paper, a gray wolf is defined as the positions of all the SPs in an SPN. The fitness of a gray wolf can be calculated by Equation (24), which represents the weighted sum of GDOP and energy balance indicators, reflecting the comprehensive performance of a gray wolf. From the analysis above, it can be seen that the goal of SPN geometry optimization is to find the SP positions that entail minimizing the objective function (24). This is equivalent to finding the gray wolf that obtains the minimum fitness. Specific steps of SPN geometry optimization can be described as follows [30].

Step 1: Initialize GWO parameters, including population size n_w, maximum iteration number n_m, and others. Set the optimal population fitness as infinite and set the iteration counter as 1.

Step 2: Initialize a gray wolf population randomly.

Step 3: For each gray wolf in the current population, check whether it meets the no co-location constraint discussed in Section 4.1. If not, modify the co-located SP positions until the no co-location constraint is met.

Step 4: For each gray wolf in the current population, check whether it meets the individual energy balance constraint discussed in Section 4.1. If not, set the fitness of the gray wolf as infinite.

Step 5: For each gray wolf to meet the constraints in Section 4.1, calculate its fitness defined in Equation (24).

Step 6: Select the minimal fitness as the optimal population fitness. Update the alpha wolf according to the gray wolf with the minimal fitness.

Step 7: Update all the gray wolves in the population with GWO strategies, such as approaching, surrounding, and attacking prey.

Step 8: For each updated gray wolf, check whether it is outside of the search space. If so, put it on the edge of the search space.

Step 9: Increase the iteration counter by 1.

Step 10: Stop the iteration if the iteration counter reaches the maximum iteration number n_m; otherwise, go to step 3 and continue the iteration.

Step 11: Iteration ends. The current alpha wolf is returned as the optimal solution, and an SPN geometry configuration can be achieved based on the optimal solution.

5. Simulation and Discussion

To verify the proposed algorithm, simulations are carried out in the scenario of SPN positioning. The simulation environment is MATLAB 2018b, and the simulation parameters are shown in Table 2. It is assumed that users are distributed in the service area uniformly with an interval of 0.3 degree, which means there are 11 users in both the longitude direction and the latitude direction.

Table 2. Simulation parameter setting.

Symbolic	Physical Meaning	Value
n_p	number of SPs in the network	6
V_{pj}	volume of stratospheric airship/m^3	100,000
C_{Dj}	aerodynamic resistance coefficient	0.027
η_{ej}	electric motor efficiency	0.7
η_{pj}	propeller propulsive efficiency	0.9
η_{vj}	photoelectric conversion efficiency	16%
P_{pj}	other equipment power/W	100
I_{SC}	the solar constant/W/m^2	1367
n_w	wolf population size	100
n_m	maximum iteration number	50
w_1	weight of GDOP in the objective function	0.5
w_2	weight of energy balance in the objective function	0.5
λ_{min}	minimum longitude/degree	90E
λ_{max}	maximum longitude/degree	93E
Φ_{min}	minimum latitude/degree	37N
Φ_{max}	maximum latitude/degree	40N
h_{min}	minimum altitude to deploy SP/km	18
h_{max}	maximum altitude to deploy SP/km	20
d_{st}	start day of the station-keeping (day number in a year)	80
d_{ed}	end day of the station-keeping (day number in a year)	80
t_{st}	start time of the station-keeping (hour in a day)	0
t_{ed}	end time of the station-keeping (hour in a day)	24
n_u	the number of observers distributed in the service area	121

5.1. Comparison of Simulations with/without Consideration of Energy Balance Requirement

To compare the proposed algorithm and traditional algorithms without consideration of the energy balance requirement, two simulations are carried out with almost the same steps and conditions as listed above, except for two differences.

The first difference is the weight assignment in the optimization object function, i.e., w_1 and w_2 in Equation (24). In the simulation without considering the energy balance requirement, w_1 is set as 1 and w_2 is set as 0. In the simulation with consideration of the energy balance requirement, both w_1 and w_2 are set as 0.5.

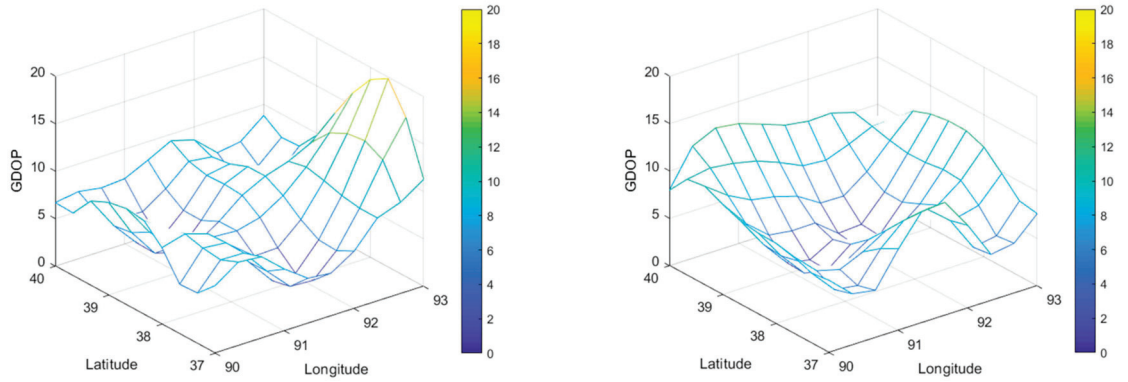
The second difference is the individual energy balance constraint. In the simulation without consideration of the energy balance requirement, the constraint is ignored, i.e., step 3 of the optimization procedure in Section 4.2 is omitted. In the simulation with consideration of the energy balance requirement, step 3 is executed.

The simulation results are listed in Table 3 and Figures 6 and 7.

In Table 3 and subsequent simulation result tables, columns λ_j, Φ_j, and h_j list the longitudes, latitudes, and altitudes of all the SPs in the SPN, column B_j lists the values of the energy balance indicators for all the SPs in the SPN, column B lists the value of the energy balance indicator for the SPN, column $GDOP_N$ lists the GDOP value for the SPN, and column F lists the fitness of the SPN. For columns B_j, B, $GDOP_N$, and F in these tables, the smaller their value is, the better the SPN geometry performance is.

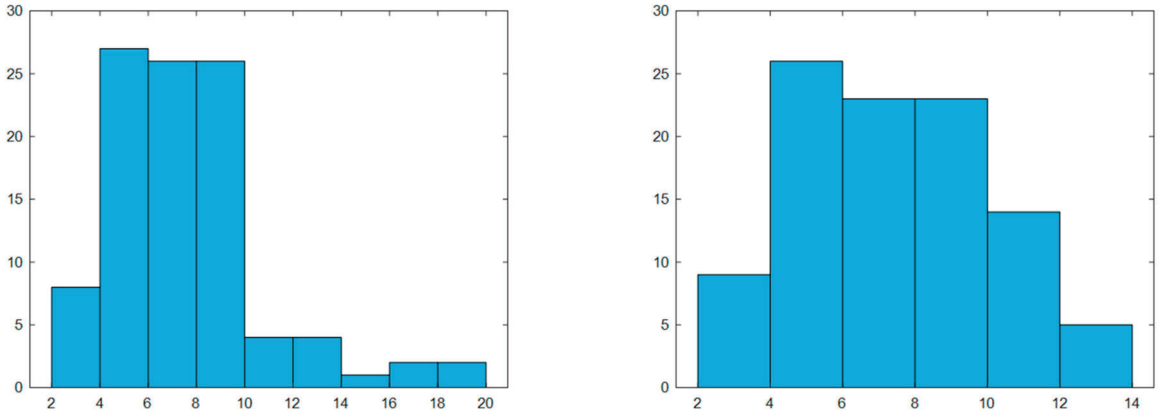
Table 3. Comparison of simulations with/without consideration of energy balance requirement.

	λ_j/deg, Φ_j/deg, h_j/km	Bj	B	$GDOP_N$	F
with consideration of energy balance requirement	92.7E, 37.0N, 19.5	0.35	1.75	7.52	4.64
	92.4E, 39.4N, 20.0	0.29			
	90.6E, 38.5N, 20.0	0.25			
	92.1E, 37.9N, 20.0	0.23			
	90.0E, 39.7N, 20.0	0.29			
	90.6E, 37.0N, 19.5	0.34			
without consideration of energy balance requirement	91.8E, 38.8N, 18.5	1.11	5.88	7.36	7.36
	90.3E, 39.4N, 18.5	1.12			
	90.3E, 37.6E, 18.5	1.03			
	91.5E, 37.9N, 19.0	0.64			
	93.0E, 37.0N, 18.0	1.67			
	93.0E, 40.0N, 20.0	0.31			



(**a**) (**b**)

Figure 6. GDOP distribution of SPNs: (**a**) SPN with consideration of energy balance requirement; (**b**) SPN without consideration of energy balance requirement.



(**a**) (**b**)

Figure 7. GDOP histogram of SPNs: (**a**) SPN with consideration of energy balance requirement; (**b**) SPN without consideration of energy balance requirement.

Column B_j in Table 3 implies that geometry configuration has a significant impact on SPN energy balance. For the same SP, their energy balance indicator can differ by several times at different locations.

Column *B* in Table 3 also proves the necessity of energy balance analysis in SPN geometry design. If it is ignored, an SPN with unacceptable energy performance might be obtained, as shown in the 7th, 8th, 9th, and 11th row of Table 3 (B_j = 1.11, 1.12, 1.03, and 1.67, respectively). The station-keeping capacity of such an SPN is very poor, which will degrade the SPN availability and continuity.

In contrast, by assigning a weight to the energy balance indicator properly in the objective function and implementing an individual energy balance constraint, the proposed algorithm can avoid unacceptable results effectively. Furthermore, in terms of GDOP, the results of the algorithm considering energy are not much worse than those of algorithms that do not consider energy.

5.2. Comparison among Different Altitude Intervals

In this section, the values of h_{min} and h_{max} in Table 2 are adjusted to compare SPN geometry performances at different altitudes. Simulations in this section and subsequent sections are implemented with consideration of the energy balance requirement.

Simulation results are shown in Table 4 and Figures 8 and 9.

Table 4. Comparison of SPNs in different altitude intervals.

Altitude Interval/km	λ_j/deg, Φ_j/deg, h_j/km	*Bj*	*B*	$GDOP_N$	*F*
18~20	92.7E, 37.0N, 19.5	0.35	1.75	7.52	4.64
	92.4E, 39.4N, 20.0	0.29			
	90.6E, 38.5N, 20.0	0.25			
	92.1E, 37.9N, 20.0	0.23			
	90.0E, 39.7N, 20.0	0.29			
	90.6E, 37.0N, 19.5	0.34			
20~22	90.9E, 38.5N, 21.0	0.10	0.52	7.12	3.82
	90.0E, 40.0N, 21.5	0.10			
	90.6E, 37.0N, 21.5	0.05			
	92.4E, 39.1N, 22.0	0.06			
	92.1E, 37.6N, 20.5	0.13			
	93.0E, 37.6N, 21.0	0.08			
22~24	92.4E, 37.9N, 22.0	0.04	0.23	6.29	3.26
	93.0E, 40.0N, 24.0	0.03			
	91.8E, 39.1N, 22.0	0.06			
	91.2E, 39.7N, 23.0	0.04			
	90.0E, 37.0N, 22.0	0.03			
	90.9E, 37.9N, 23.0	0.03			
24~26	90.9E, 38.2N, 26.0	0.01	0.11	5.58	2.84
	93.0E, 40.0N, 26.0	0.02			
	92.1E, 37.6N, 25.0	0.01			
	90.0E, 39.7N, 25.0	0.03			
	90.0E, 37.0N, 24.0	0.02			
	92.1E, 39.1N, 25.5	0.02			
26~28	93.0E, 40.0N, 28.0	0.01	0.06	5.35	2.71
	90.6E, 39.7N, 27.5	0.01			
	92.4E, 37.6N, 26.5	0.01			
	90.0E, 37.0N, 26.0	0.01			
	92.1E, 39.1N, 28.0	0.01			
	90.6E, 37.9N, 27.5	0.01			

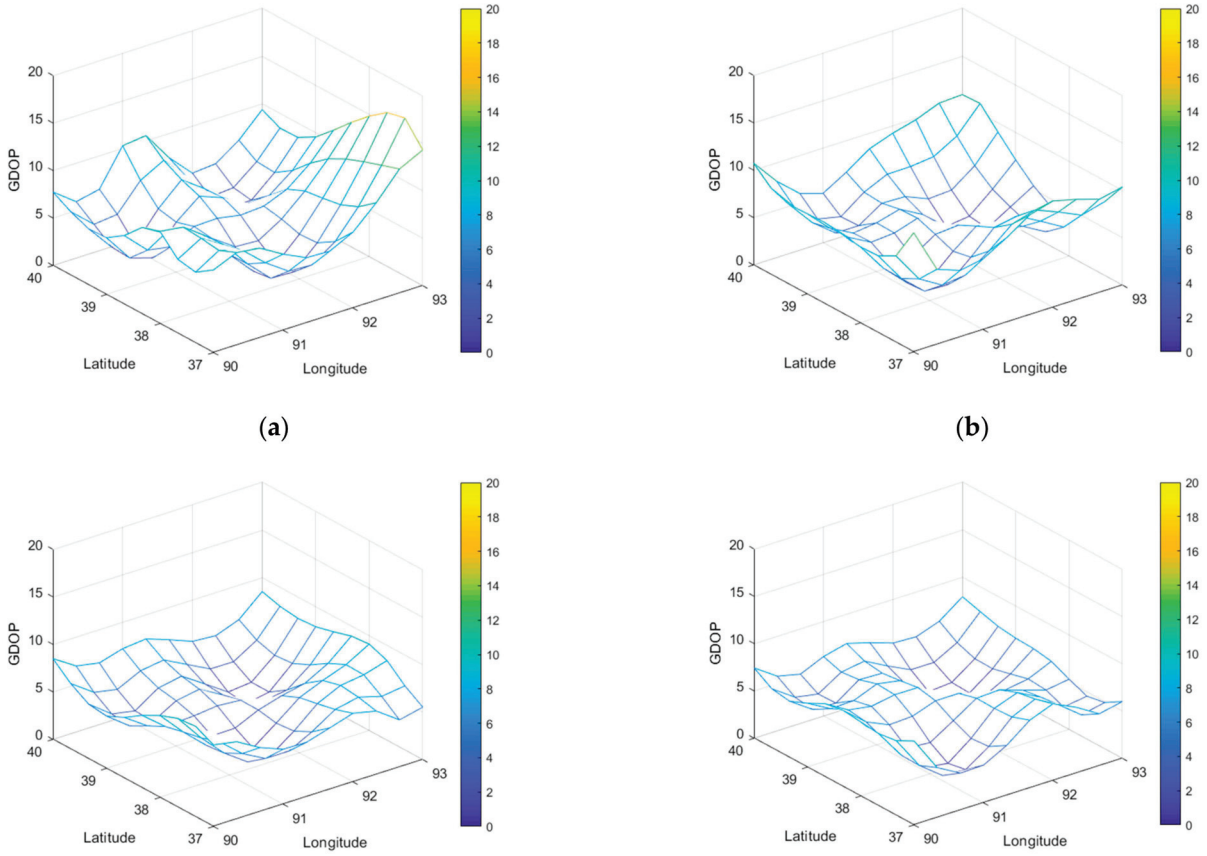


Figure 8. GDOP distribution of SPNs in different altitude intervals: (**a**) SPNs in 20~22 km; (**b**) SPNs in 22~24 km; (**c**) SPNs in 24~26 km; (**d**) SPNs in 26~28 km (for GDOP distribution of SPNs in 18~20 km, please refer to Figure 6a).

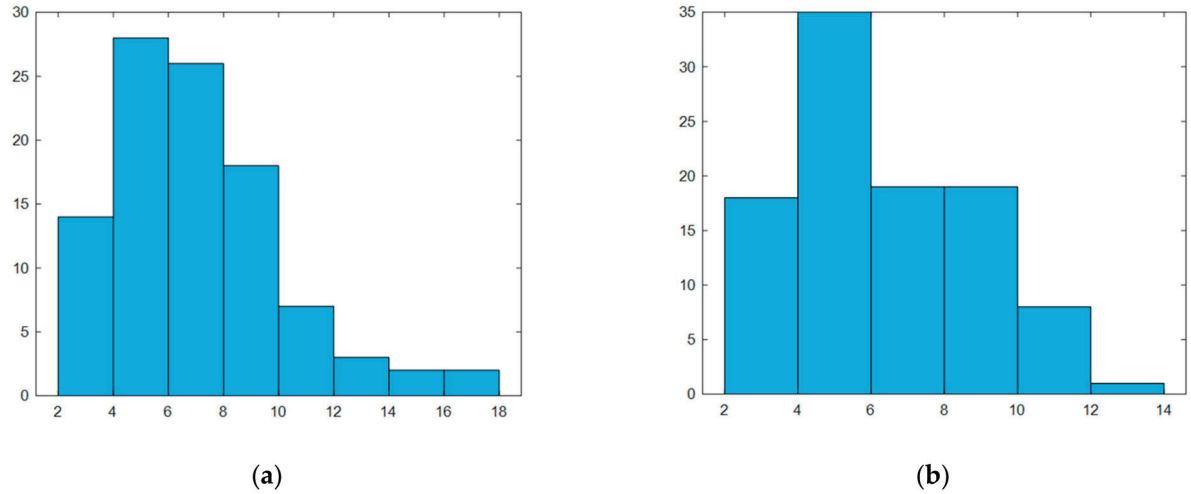
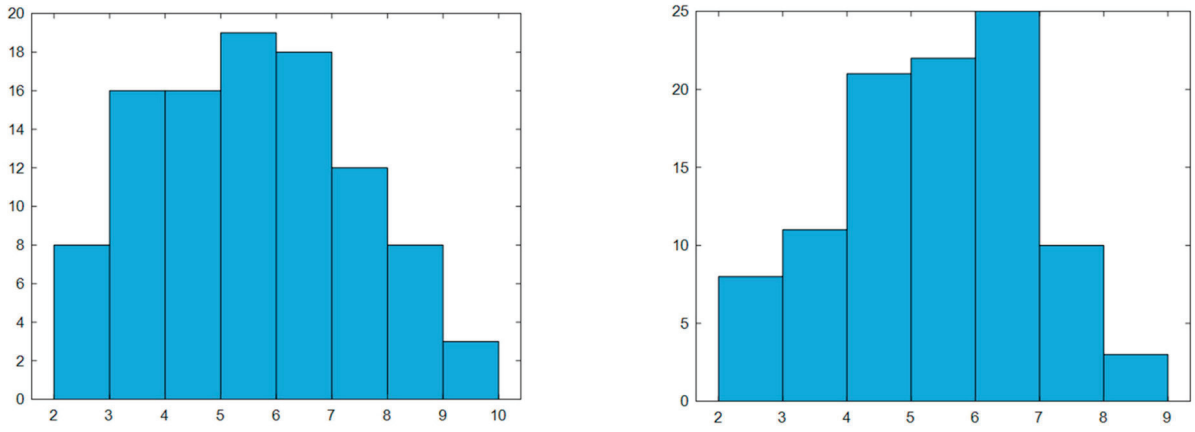


Figure 9. *Cont*.



(**c**) (**d**)

Figure 9. GDOP histogram of SPNs in different altitude intervals: (**a**) SPNs in 20~22 km; (**b**) SPNs in 22~24 km; (**c**) SPNs in 24~26 km; (**d**) SPNs in 26~28 km (for GDOP histogram of SPNs in 18~20 km, please refer to Figure 7a).

The results in Table 4 and Figures 8 and 9 imply that under the given conditions, both GDOP and the energy balance indicator show a decreasing trend with the altitude increasing from 18 km to 28 km, meaning that SPN geometry performance improves as altitude increases in this altitude interval.

The decrease in the energy balance indicator can be attributed to wind, atmosphere, solar radiation, and other factors.

From Figure 3, it can be seen that in March, as the altitude increases from 18 km to 28 km, the zonal wind speed decreases from over 20 m/s to less than 10 m/s, while the increase in the meridional wind speed is less than 5 m/s, resulting in a decrease in energy consumption for the SP. From Figure 4, it can be seen that as the altitude increases, the atmosphere density decreases, which is also beneficial for reducing energy consumption.

From Figure 10, it can be seen that the atmosphere pressure decreases as the altitude increases. From Equations (13)–(15), it can be inferred that solar direct radiation intensity increases as the altitude increases, which can lead to an increase in SP energy production, as illustrated in Figure 5.

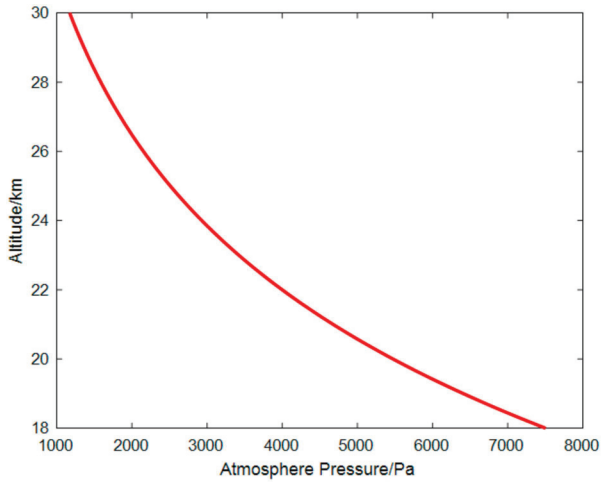


Figure 10. Atmosphere pressure from 18 km to 30 km according to U.S. Standard Atmosphere, 1976 [24].

Therefore, the energy balance indicator, which is the ratio of energy consumption and energy production, decreases as the altitude increases.

In addition, as the altitude increases, the value of the SPN elevation angle increases, which can improve the GDOP of the SPN. This has been deduced and explained in detail in [8,10].

Therefore, both the energy balance indicator and GDOP will improve as the altitude increases, as shown in Table 4, Figure 8, and Figure 9. So, an SPN with a higher station-keeping altitude is expected in order to achieve a better performance under the given conditions.

5.3. Comparison among Different Latitude Intervals

In this section, the values of Φ_{min} and Φ_{max} in Table 2 are adjusted to compare network performances in different latitudes. Simulation results are shown in Table 5.

Table 5. Comparison among SPNs of different latitude intervals.

Latitude Interval/deg	λ_j/deg, Φ_j/deg, h_j/km	Bj	B	$GDOP_N$	F
33~36N	91.5E, 35.1N, 20.0	0.15	1.30	7.15	4.23
	92.4E, 33.0N, 19.5	0.20			
	93.0E, 36.0N, 20.0	0.17			
	90.0E, 34.2N, 20.0	0.12			
	90.0E, 36.0N, 19.0	0.54			
	91.8E, 33.9N, 20.0	0.12			
35~38N	90.6E, 35.3N, 19.5	0.28	1.61	7.32	4.47
	92.4E, 35.3N, 19.0	0.52			
	92.1E, 36.8N, 20.0	0.19			
	93.0E, 38.0N, 20.0	0.23			
	91.2E, 36.2N, 20.0	0.17			
	91.2E, 37.7N, 20.0	0.22			
37~40N	92.7E, 37.0N, 19.5	0.35	1.75	7.52	4.64
	92.4E, 39.4N, 20.0	0.29			
	90.6E, 38.5N, 20.0	0.25			
	92.1E, 37.9N, 20.0	0.23			
	90.0E, 39.7N, 20.0	0.29			
	90.6E, 37.0N, 19.5	0.34			
39~42N	92.1E, 41.4N, 19.5	0.55	2.55	7.12	4.83
	90.9E, 39.0N, 20.0	0.27			
	93.0E, 42.0N, 20.0	0.40			
	90.9E, 39.9N, 19.5	0.47			
	92.1E, 40.2N, 20.0	0.32			
	90.3E, 41.4N, 19.5	0.54			
41~44N	90.9E, 41.9N, 20.0	0.39	3.14	6.89	5.02
	93.0E, 44.0N, 20.0	0.49			
	90.0E, 41.0N, 20.0	0.35			
	90.3E, 42.8N, 20.0	0.43			
	91.8E, 43.1N, 19.5	0.63			
	92.4E, 42.2N, 19.0	0.85			

The results detailed in Table 5 imply that low latitude intervals tend to be beneficial to the network energy balance while having little impact on SPN GDOP under the given conditions.

From Figure 2, it can be seen that within the service area listed in Table 5, as the latitude decreases, the zonal wind speed decreases significantly, which can reduce SP energy consumption. From Figure 5, it can be seen that lower latitude can help SPs obtain more energy production. Therefore, the SPN energy balance indicator shows a decreasing tendency with decreasing latitude.

6. Conclusions and Future Works

SPN is a novel aerial network with promising potential. Geometry design is a critical problem affecting its service performance significantly. This paper focuses on SPN geometry design to pursue a satisfactory performance for both GDOP and energy balance. In the assumed service area and under the given simulation conditions, the following conclusions can be drawn:

Geometry configuration has a significant impact on both SPN energy balance and GDOP. Consequently, neither of them can be ignored in SPN geometry design.

The energy balance requirement of an SPN can be met by properly assigning weights on the energy balance indicator in the objective function and implementing the energy balance constraint of individual SPs.

Both GDOP and energy balance can be improved by raising the station-keeping altitude towards the altitude interval of 26~28 km.

GDOP shows no substantial improvement when the deployment space is slightly adjusted southward and northward. Energy balance tends to improve gently when deployment space moves southward.

Some issues can be analyzed in the future.

In this paper, the photoelectric conversion efficiency of the solar array is assumed to be constant. However, it changes with the thermal conditions in practice. In the future, the influence of photoelectric conversion efficiency fluctuation on SP energy production and energy balance should be analyzed.

Also, uncertain wind is ignored in this paper since the wind at the altitude interval of SP is relatively stable, but uncertain wind exists according to [32], and it may have an impact on the energy consumption of SPs. Further analysis can be conducted next.

In addition, this paper implements simulations currently just for a small area. More simulations for larger areas can be carried out according to requirements.

Author Contributions: Conceptualization, Y.Q.; methodology, Y.Q. and S.W.; software, Y.Q.; validation, H.F. and Q.L.; writing—original draft preparation, Y.Q.; writing—review and editing, S.W. and H.F.; visualization, Y.Q.; supervision, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key Research and Development Program of China (Grant No. 2022YFB3901805).

Data Availability Statement: Data are contained within the article.

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

References

1. Tsujii, T.; Rizos, C.; Wang, J.; Dai, L.; Roberts, C. A navigation/positioning service based on pseudolites installed on stratospheric airships. *Jpn. Soc. Aeronaut. Space Sci.* **2003**, *50*, 36. [CrossRef]
2. Dovis, F.; Presti, L.; Mulassano, P. Support infrastructures based on high altitude platforms for navigation satellite systems. *IEEE Wirel. Commun.* **2005**, *12*, 106–112. [CrossRef]
3. Zheng, H.; Atia, M.; Yanikomeroglu, H. High Altitude Platform Station (HAPS)-Aided GNSS for Urban Areas. *arXiv* **2023**. [CrossRef]
4. Chandu, B.; Pant, R.S.; Moudgalya, K. Modeling and simulation of a precision navigation system using pseudolites mounted on airships. In Proceedings of the 7th AIAA ATIO Conf, 2nd CEIAT Int'l Conf on Innov and Integr in Aero Sciences, 17th LTA Systems Tech Conf, Belfast, UK, 18–20 September 2007. [CrossRef]
5. Dai, L.W.; Wang, J.L.; Tsujii, T.; Rizos, C. Pseudolite applications in positioning and navigation: Modelling and geometric analysis. *J. Navig.* **2004**. Available online: https://www.researchgate.net/profile/Jinling_Wang2/publication/2865922_Pseudolite_applications_in_positioning_and_navigation_Modelling_and/links/546d98510cf26e95bc3cb846.pdf?__cf_chl_tk=NFi8SdAKc5gBoKKe9i2BXt_BS1bZjkE2X5PooWMY56U-1717999627-0.0.1.1-8446 (accessed on 4 June 2024).
6. Fateev, Y.L.; Ratuschnyak, V.N.; Kartson, I.N.; Tyapkin, V.N.; Dmitriev, D.D.; Goncharov, A.E. Analyzing measurement errors for navigation parameters in onground short-range navigation systems based on pseudolites. *IOP Conf.* **2016**, *155*, 012016. [CrossRef]
7. Sang, W.G.; He, X.F.; Chen, Y.Q. Configuration of pseudolite-alone positioning system based on DOP geometry structure. *Bull. Surv. Mapp.* **2013**, *9*, 1–4.
8. Hu, W.; Yang, J.J.; He, P. Study on pseudolite configuration scheme based on near space airships. *Radio Eng.* **2009**, *39*, 24–27.

9. Gao, S.S.; Zhao, F.; Xie, M.L. Research on the geometric configuration scheme of near space pseudolite-only positioning system. *J. Navig. Position.* **2013**, *1*, 21–25.
10. Yang, Y.; Gao, S.S.; Yan, H.F. Design on geometric configuration schemes of pseudolite in near space. *Syst. Eng. Electron.* **2014**, *36*, 532–538.
11. Mosavi, M.R.; Divband, M. Calculation of geometric dilution of precision using adaptive filtering technique based on evolutionary algorithms. In Proceedings of the 2010 International Conference on Electrical and Control Engineering, Wuhan, China, 25–27 June 2010. [CrossRef]
12. Shao, K.; Li, K.; Wang, J. PSO-Based pseudolite layout strategy. *Commun. Technol.* **2017**, *50*, 2454–2459.
13. Tang, W.J.; Chen, J.P.; Yu, C.; Ding, J.; Wang, R. A new ground-based pseudolite system deployment algorithm based on MOPSO. *Sensors* **2021**, *21*, 5364. [CrossRef]
14. Yang, L.; Zhou, J.H.; Chen, J.P. The study of optimization of formation flying navigation augmentation platforms based on genetic algorithm. *GNSS World China* **2008**, *33*, 9–13.
15. Chen, C.; Chen, K.; Huang, J.; Li, Y. Using genetic algorithms to approximate weighted geometric dilution of precision. In Proceedings of the 2016 International Symposium on Computer, Consumer and Control (IS3C), Xi'an, China, 4–6 July 2016. [CrossRef]
16. Song, J.; Hou, C.; Xue, G.; Ma, M. Study of constellation design of pseudolites based on improved adaptive genetic algorithm. *J. Commun.* **2016**, *11*, 879–885. [CrossRef]
17. Araripe, D.F.; De, M.F.; Campos, D.T. High-altitude platforms—Present situation and technology trends. *J. Aerosp. Technol. Manag.* **2016**, *8*, 249–262. [CrossRef]
18. Nickol, C.L.; Guynn, M.D.; Kohout, L.L.; Ozoroski, T.A. High Altitude Long Endurance UAV Analysis of Alternatives and Technology Requirements Development. In Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 8–11 January 2007. [CrossRef]
19. Yang, X.; Wang, F.; Liu, W.; Xiao, W.; Ye, X. A Layout Method of Space-Based Pseudolite System Based on GDOP Geometry. *Chin. J. Electron.* **2023**, *32*, 1050–1058. [CrossRef]
20. Sultana, Q.; Sunehra, D.; Srinivas, V.S.; Sarma, A.D. Effects of Pseudolite Positioning on DOP in LAAS. *Positioning* **2010**, *1*, 18–26. [CrossRef]
21. Jiang, M.; Li, R.; Liu, W. Research on geometric configuration of pseudolite positioning system. *Comput. Eng. Appl.* **2017**, *53*, 271–276.
22. Zhao, Y.; Garrard, W.; Mueller, J. Benefits of Trajectory Optimization in Airship Flights. In Proceedings of the AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit, Chicago, IL, USA, 20–23 September 2004. [CrossRef]
23. Wu, L.; Li, Y.; Li, Z. The research of route planning for stratospheric airships based on genetic algorithms. *Spacecr. Recovery Remote Sens.* **2011**, *32*, 1–6.
24. *NOAA-S/T-76-1562*; U.S. Standard Atmosphere, 1976. National Oceanic and Atmospheric Administration; National Aeronautics and Space Administration; United States Air Force: Washington, DC, USA, 1976.
25. Stanciu, C.; Stanciu, D. Optimum tilt angle for flat plate collectors all over the World—A declination dependence formula and comparisons of three solar radiation models. *Energy Convers. Manag.* **2014**, *81*, 133–143. [CrossRef]
26. Zhang, L.; Li, J.; Jiang, Y.; Du, H.; Zhu, W.; Lv, M. Stratospheric airship endurance strategy analysis based on energy optimization. *Aerosp. Sci. Technol.* **2020**, *100*, 105794. [CrossRef]
27. Long, Y.; Deng, X.; Yang, X.; Hou, Z. Trajectory simulation of stratosphere aerostats in polar vortex wind field. *Comput. Simul.* **2021**, *38*, 37–42.
28. Lv, M.; Li, J.; Du, H.; Zhu, W.; Meng, J. Solar array layout optimization for stratospheric airships using numerical method. *Energy Convers. Manag.* **2017**, *135*, 160–169. [CrossRef]
29. Zhang, Y.; Li, J.; Lv, M.; Tan, D.; Zhu, W.; Sun, K. Simplified analytical model for investigating the output power of solar array on stratospheric airship. *Int. J. Aeronaut. Space Sci.* **2016**, *17*, 432–441. [CrossRef]
30. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
31. Arora, S.; Singh, H.; Sharma, M.; Sharma, A.; Anand, P. A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. *IEEE Access* **2019**, *7*, 26343–26361. [CrossRef]
32. Conway, J.P.; Bodeker, G.E.; Waugh, D.W.; Murphy, D.J.; Cameron, C.; Lewis, J. Using project Loon superpressure balloon observations to investigate the inertial peak in the intrinsic wind spectrum in the midlattitude stratosphere. *J. Geophys. Res. Atmos.* **2019**, *124*, 8594–8604. [CrossRef]

electronics

MDPI

Article

FLsM: Fuzzy Localization of Image Scenes Based on Large Models

Weiyi Chen [1,*], Lingjuan Miao [1], Jinchao Gui [2], Yuhao Wang [1] and Yiran Li [1]

[1] School of Automation, Beijing Institute of Technology, Beijing 100081, China; miaolingjuan@bit.edu.cn (L.M.)
[2] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100101, China; guijinchao@aircas.ac.cn
* Correspondence: 3220195102@bit.edu.cn

Abstract: This article primarily focuses on the study of image-based localization technology. While traditional methods have made significant advancements in technology and applications, the emerging field of visual image-based localization technology demonstrates tremendous potential for research. Deep learning has exhibited a strong performance in image processing, particularly in developing visual navigation and localization techniques using large-scale visual models. This paper introduces a sophisticated scene image localization technique based on large models in a vast spatial sample environment. The study involved training convolutional neural networks using millions of geographically labeled images, extracting image position information using large model algorithms, and collecting sample data under various conditions in elastic scene space. Through visual computation, the shooting position of photos was inferred to obtain the approximate position information of users. This method utilizes geographic location information to classify images and combines it with landmarks, natural features, and architectural styles to determine their locations. The experimental results show variations in positioning accuracy among different models, with the most optimal model obtained through training on a large-scale dataset. They also indicate that the positioning error in urban street-based images is relatively small, whereas the positioning effect in outdoor and local scenes, especially in large-scale spatial environments, is limited. This suggests that the location information of users can be effectively determined through the utilization of geographic data, to classify images and incorporate landmarks, natural features, and architectural styles. The study's experimentation indicates the variation in positioning accuracy among different models, highlighting the significance of training on a large-scale dataset for optimal results. Furthermore, it highlights the contrasting impact on urban street-based images versus outdoor and local scenes in large-scale spatial environments.

Keywords: localization of image scenes; fuzzy localization; large models; image processing; deep learning

1. Introduction

In recent years, the Beidou Satellite Navigation System (BDS), Galileo Satellite Navigation System (Galileo), modern Global Navigation System (GPS), and Global Navigation System (GLONASS) have been developed. There are over 140 GNSS satellites available [1]. Global Navigation Satellite Systems (GNSSs) are increasingly used for outdoor navigation [2]. The strap-on inertial navigation system (SINS) can automatically measure the user's position, speed, and attitude [3]. However, the inertial navigation system is subject to its own limitations and high costs. Its primary drawback is the increase in error over time, leading to drift. The aforementioned systems represent traditional navigation and positioning technologies. Over years of development, these technical systems have essentially established a relatively comprehensive technical framework, which is extensively utilized in human activities and daily life. As human science and technology advance, the need for navigation and positioning continues to evolve. Robust visual localization over

long periods of time is one of the biggest challenges for the long-term navigation of mobile robots [4]. Monocular visual inertial navigation systems (VINSs) are widely used in fields such as robot navigation, autonomous driving, and augmented/virtual reality [5]. The current emergence of various intelligent robots not only significantly facilitates our lives, but also presents higher performance requirements.

Visual navigation is a navigation method using visible and invisible imaging technology, which has the advantages of good concealment, strong autonomy, fast and accurate measurement, low cost, and high reliability [6]. At present, the emergence of diverse intelligent robots not only significantly facilitates human life, but also presents elevated demands for robot performance, thereby establishing a prerequisite for the advancement of visual positioning technology. Visual information is increasingly utilized in navigation applications. With the introduction of numerous new concepts, methods, and theories, image processing technology based on deep learning has progressively matured. Visual navigation technology is expected to be developed and widely used in the fields of aircraft, unmanned aerial vehicles, various cruise missiles, deep space probes, indoor mobile navigation, and so on [7]. Therefore, visual navigation technology has high research and application value in the field of navigation. Usually, visual navigation on robots is achieved by installing a monocular or binocular camera to obtain local images of the environment and to make navigation decisions. The research on intelligent robots began in the late 1960s, marked by Shakey, the first mobile robot developed by Stanford Research Institute (SRI) [8]. Its main objective is to study the real-time control of robot systems in complex environments. Representative examples include urban robots and tactical robots developed by Jet Propulsion Laboratory [9–11]. These robots are equipped with binocular stereo vision systems for obstacle detection. Visual navigation and positioning can also be applied to spacecraft or interplanetary detectors, such as lunar probes. The lunar rover has a high degree of autonomy and is suitable for performing exploration tasks in a complex and unstructured lunar environment [12–15]. The stereo vision system of the lunar rover is the most direct and effective tool for close-range and high-altitude moon detection. It serves as a tool for understanding the lunar environment and provides crucial information for lunar rover survival in complex environments. Using the stereo vision system, we can not only reconstruct the terrain of the environment in real time to avoid obstacles, but also use the obtained stereo sequence images to estimate the movement of the rover itself. Therefore, the application of visual navigation in the field of robotics is extremely extensive and significant [16–18].

The essence of visual navigation is to obtain the two-dimensional image information of the scene through one or more cameras, and then determine the operation information of navigation by using image processing, computer vision, pattern recognition, and other algorithms. The techniques involved include camera calibration, stereo image matching, path identification, and 3D reconstruction [19,20]. Inspired by the process of robot positioning and navigation, we contemplate the extraction of positional data from images and the potential for a single image to facilitate robot navigation and positioning tasks. Addressing this challenge, we have conducted extensive research. Our primary focus is on the feasibility and precision of obtaining location information from images. If it is possible to probabilistically infer geographic location data from images, it would constitute a highly significant area of study. Visual positioning can play an important role in satellite navigation failures and has a broad application value. When satellite navigation signals are obstructed or unavailable, visual positioning systems can provide reliable positioning information, suitable for many fields, such as autonomous driving and unmanned vehicles. With the help of high-precision visual positioning systems, autonomous vehicles can accurately locate and navigate in environments without satellite signals, ensuring safe driving in cities. Indoor navigation—in indoor environments, satellite signals are usually weak or unavailable. By using visual positioning technology, people can accurately navigate and locate within large buildings such as shopping malls, airports, and hospitals. Industrial robots—in the fields of manufacturing and logistics, industrial robots require

precise positioning information to perform various tasks. Visual positioning systems can provide real-time location information to help robots accurately perform tasks. Search and rescue—in the event of a disaster, satellite navigation systems may be disrupted or damaged. Visual positioning technology can help search and rescue personnel locate trapped individuals, without satellite signals. Military applications—military departments can use visual positioning systems for precise positioning and navigation, without being disturbed or monitored by hostile forces. Mobile devices—smartphones and tablets can use cameras and sensors for visual positioning, providing users with indoor navigation, augmented reality experiences, and other functions.

This paper focuses on geographical fuzzy positioning using image information mining, consolidates the relevant technical accomplishments in navigation and positioning, and scrutinizes the limitations of current navigation and positioning technology based on their characteristics. Addressing the requirements of visual navigation and positioning, this paper aims to achieve visual positioning capability in a lightweight manner for various scenarios and applications. It introduces a method of image fuzzy positioning based on a large model, enabling scene location determination through images in GPS failure environments. A schematic diagram of FLsM, based on image localization, is shown in Figure 1. The key contributions of this paper are summarized as follows:

- The concept of elastic scale space is introduced, which refers to a coupling space between large-scale scenes and fine scenes, emphasizing the variability and unpredictability of the environment.
- A vision-based fuzzy positioning technology is suggested, emphasizing the semantic information extraction from the visual image itself and providing geographic location.
- By leveraging multiple models for image training, employing advanced deep learning models, and utilizing a large dataset of Internet data for pre-training, we can efficiently match images and texts and accomplish the fuzzy positioning of images.

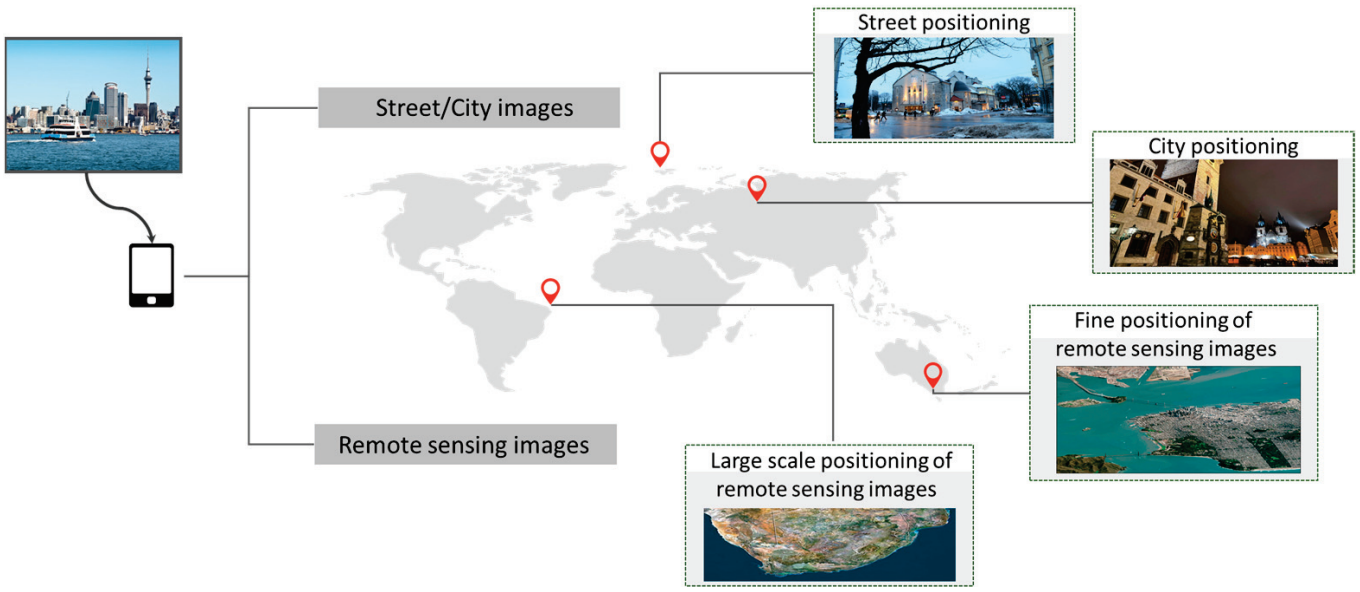


Figure 1. Schematic diagram of image fuzzy positioning.

2. Related Works

In this section, we introduce the related technologies of visual navigation and positioning. Including traditional slam technology, deep learning models, geographic information positioning research, and so on.

2.1. Traditional SLAM Technology

The traditional craft of Simultaneous Localization and Mapping (SLAM) techniques has been a beacon in the odyssey of robotic navigation and mapping [21]. These methods typically harness the power of sensor fusion, utilizing data from cameras, lidars, and other sensors to simultaneously decipher the robot's location and construct a map of its environment. Algorithms such as ORB-SLAM2 and ORB-SLAM3 have been luminaries in this domain, demonstrating significant advancements in accuracy and stability over the past decade [22]. However, as underscored in recent scholarly pursuits, challenges persist in achieving long-term robustness, especially in the face of diverse and dynamic environmental perturbations.

2.2. Deep Learning Models for Visual Navigation

The advent of deep learning has revolutionized the landscape of visual navigation [23,24]. Contemporary approaches leverage the prowess of neural networks to learn complex mappings between visual inputs and navigational outputs. CLIP, as discussed in the previous section, emerges as a model capable of learning transferable visual representations from the vast expanse of natural language supervision [25,26]. It extends the scope of computer vision systems by directly learning from the raw text about images, showcasing promising results across a myriad of tasks, without the need for task-specific training.

Moreover, recent scholarly endeavors delve into the comparison of open-source visual SLAM approaches, evaluating algorithms based on factors such as accuracy, computational performance, and robustness. This reflects the ongoing quest to enhance the performance of visual navigation systems and address the specific challenges posed by different scenarios and datasets.

2.3. Geographic Information Positioning Research

Geographic Information Systems (GISs) and positioning technologies are the compass and sextant of modern navigation systems. Recent works, such as GeoCLIP, integrate CLIP-inspired techniques to chart the course for effective worldwide geo-localization. By encoding GPS information and employing hierarchical learning [27,28], GeoCLIP demonstrates a state-of-the-art performance, navigating the challenges associated with the diversity of global landscapes. This underscores the importance of geographic information in refining the accuracy and reliability of visual navigation systems.

3. Materials and Methods

3.1. Overview

A visual fuzzy positioning method in elastic scale space is proposed for different application requirements of various scenes, based on an in-depth analysis of image information. This method differs from traditional visual positioning methods. High-resolution remote sensing images are used for photogrammetry in large-scale scenes to generate image maps of different scales. Fine-scale scenes are divided into indoor and outdoor areas, and environmental image data are collected from natural target sample data using mobile measuring equipment. The expectation is that these two types of data rely solely on the information carried by the image itself, and the positioning requirements can be fulfilled using deep learning algorithms with large models. The concept of elastic scale space involves leveraging the randomness and lightness of the image, focusing on mining the information value of the image and obtaining rough positioning information. The significance of this work lies in its lightweight design, which does not impose strict requirements on the image itself and emphasizes model training. The image captured by the terminal's camera is used to determine the inclusion of a specific target and then the user's precise position is calculated visually. The entire process is shown in Figure 2.

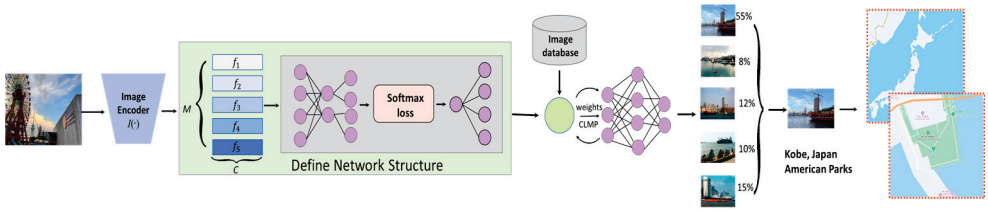


Figure 2. Flow description of image positioning algorithm.

3.2. Constructing an Elastic Scale Spatial Environment

We propose the concept of an elastic scale space, which is a space between the coupling of large-scale and fine-grained scenes, highlighting the arbitrariness and randomness of the environment, as shown in Figure 3.

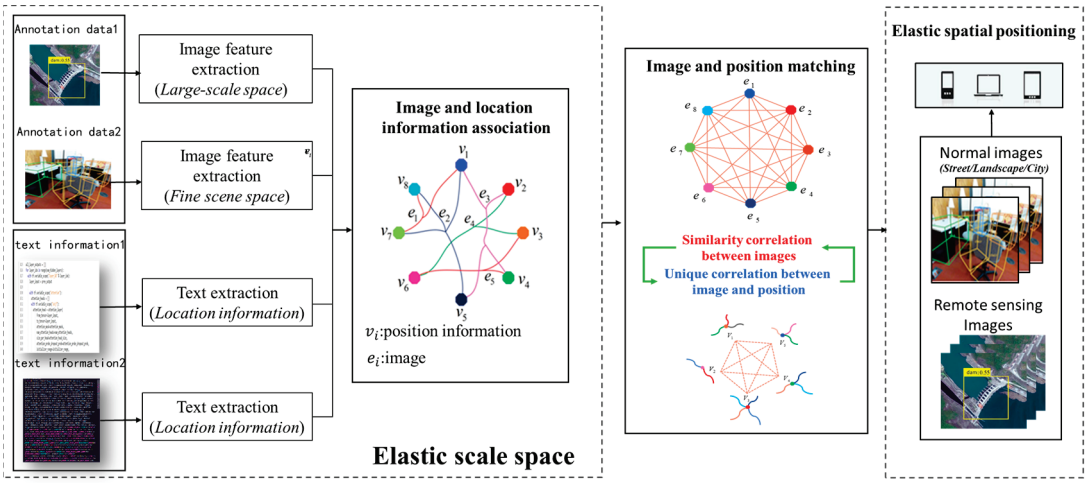


Figure 3. Overall framework of elastic spatial positioning.

Traditionally, in large-scale scenes, it is necessary to make large-scale scene image base maps and develop service engines. The research process is as follows:

- Select an appropriate reference point

The choice of reference points should be clearly defined as a clear landmark, such as road intersections, building corners, etc., that can be considered as optional reference points. After selecting the datum point, high-precision surveying and mapping should be carried out on the selected datum point.

- Image map of construction environment

Utilize aviation professional equipment (drones and aerial photography planes) for photogrammetry work, collect high-resolution remote sensing images of the environmental field, and produce image maps at scales of 1:1000, 1:500, etc.

- Unified spatiotemporal benchmark construction

The determination of the spatiotemporal baseline of the environmental field usually adopts the WEB Mercator projection method, whose core is the transformation of the coordinate system, mainly the transformation of the image into a plane coordinate system, which should be consistent with the current general map projection.

For the construction of fine scenes, the following process is required.

- Target sample collection and data construction

In order to more accurately represent environmental information, it is divided into two parts—indoor environment and outdoor environment. The outdoor environment dataset utilizes satellite positioning technology to collect and store the location information

of identifiable targets in the scene. At the same time, the mobile terminal is equipped with a high-definition camera to capture the target from various angles and to collect image information of the target. Then, utilize onboard or airborne measurement systems to collect local RGBD information around the target and construct a data resource lake for the scene through various technical means. Similarly, in indoor environments, it is necessary to establish a unified indoor coordinate system. The camera is used to capture the target from various angles, collect the image information of the target, collect the location information of identifiable targets indoors, and save it.

- Model library construction

According to the requirements, use a deep learning framework to train the collected image information, obtain a proprietary model library, and obtain parameter models that meet the requirements.

3.3. The Concept of Visual Blur Localization

In order to achieve precise positioning, this article suggests a vision-based fuzzy positioning technology that is integrated with satellite navigation and other techniques. In order to categorize or detect input scene photographs, visual blur localization focuses on mining the semantic information included in the visual images themselves, supplying positional range information, and utilizing deep learning network methods [29] in conjunction with large-scale model structures. Semantic segmentation, which effectively pulls information from the scene, is the main focus. The marked position data is extracted from the saved position data, based on the recognition findings. It should be noted that the input for future precise positioning can come from fuzzy positioning data. Transform the fuzzy position information that was previously acquired into fine scene data. Extract the target's local RGBD information from the location data that have been saved and compare it with the scene map based on the recognition findings. Use the 2D–3D visual solving algorithm to obtain the user's precise position. In Figure 4, the technical procedure is displayed.

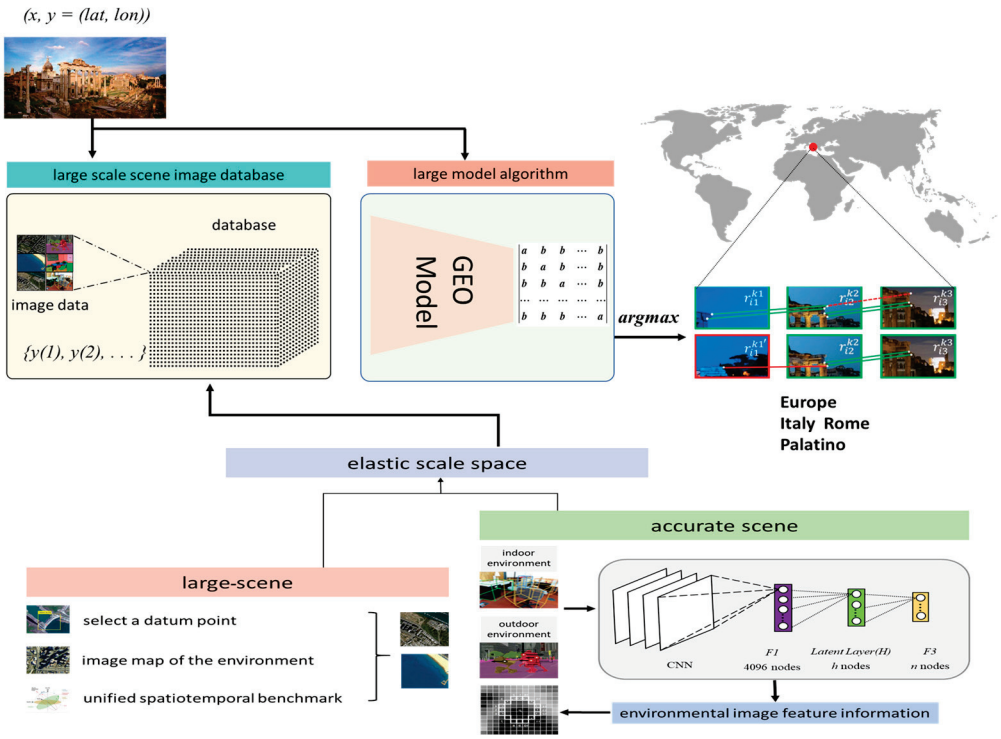


Figure 4. Visual fuzzy positioning process based on large model.

3.4. Multi-Source Image Data Source Matching

Picture retrieval and matching algorithms are essential for the search, matching, and display of location information derived from visual images. These techniques enable the rapid matching of feature data and corresponding picture data. In addition to image encoding and quick image retrieval for large amounts of data, there are numerous important technologies that still need to be resolved. These include fast feature extraction technology for multi-source image data, unified spatiotemporal benchmarks, and image encoding. Aerial surveys, other sensor image data, and satellite remote sensing photos are some of the multi-source image data used to create traditional large-scale scenes. Data integration is based on the unification of spatiotemporal benchmarks and the conversion of multi-source picture data to a common scale. The CGCS2000 coordinate system is the source of the coordinate system [30], terrain feature points, etc., in the image; it projects the image data and feature data in a plane according to a universal map. Uniformly project onto the WEB Magic Card to form a consistent spatiotemporal baseline. Complete feature extraction and spatiotemporal matching processing of multi-source image data.

3.5. Building a Large-Scale Complex Scene Graph Database

This project uses a YFCC100M dataset to obtain metadata containing Geo information, and combines scene image data such as SUN2010, Places2, and Google StreetView to construct a large-scale complex scene graph database [31–33]. Firstly, the YFCC100M metadata is processed to extract data with geo labels from the original dataset. Then, the Geo labeled dataset is transformed using the GEOPY tool to obtain an image dataset with actual location information. Utilize SIFT for feature extraction on scene datasets such as SUN2010, Places2, and Google StreetView to obtain a large database of scene images. YFCC100M obtains a text data document database after data preprocessing, which can be used as a data source for future model training. In particular, this text data contains the Geo information of each image, which is crucial for accurate scene positioning. SUN2010, Places2, and Google StreetView have undergone SIFT algorithm feature processing to obtain a relatively rich set of scene-based feature datasets, providing a reliable data source for training scene recognition models. There is an urgent need for rapid detection, autonomous warning, information confrontation, and on-site disposal of remote targets. The YFCC100M database is an imaging database that has been based on Yahoo Flickr since 2014. The library consists of 100 million pieces of media data generated between 2004 and 2014, including 99.2 million photo data and 800,000 video data [31]. The YFCC100M dataset does not contain photo or video data and each row in the document contains metadata for a photo or video. Among them, Photos/video identifiers, Longitude, and Latitude are used. Geo information refers to geographic location information, which can record the geographic location information at the time of photo shooting, namely longitude and latitude. But not all metadata contains Geo information, so it is necessary to filter out metadata that does not contain Geo information. Then, use the geopy toolkit to convert longitude and latitude to actual addresses. It is easy to obtain the geographic coordinates of a street address, city, country, and land parcel worldwide using geopy, and to parse them through third-party geoencoders and data sources.

3.6. Data Feature Extraction

The CLIP (Contrastive Language–Image Pre-Training) model is used to extract features from the YFCC100M, im2gps3k, and Google BigEarthNet datasets [34–36]. The CLIP model consists of two parts, a visual encoder and a text encoder, in which the visual encoder is used to process image information and the text encoder is used to process the address position information after reverse geocoding. The features extracted from the CLIP model have many advantages. First of all, because the CLIP model can handle both images and texts, it can understand and make use of the association between images and texts, thus extracting richer and more representative features. Secondly, the characteristics of the CLIP model are highly robust, and can remain stable even in the face of various changes (such as

illumination changes, visual angle changes, etc.). In addition, the features extracted from the CLIP model have a good generalization ability and can be applied to various tasks and scenes. Finally, the features of the CLIP model have a high degree of discrimination, which can effectively distinguish different objects and scenes. These advantages make the CLIP model perform well in various visual and language tasks. CLIP's visual encoder and text encoder are its core components. The visual encoder is responsible for extracting features from images, while the text encoder is responsible for extracting features from texts. These two encoders can extract the features of text and image, respectively, and then calculate the similarity between the text vector and the image vector to predict whether they match. This design enables the CLIP model to process both text and image at the same time, thus achieving the joint understanding of image and text. This is a major feature of the CLIP model and it is also the key to its outstanding performance in various tasks. This has provided strong support for our work.

3.7. Design of CNN-Based Visual Scene Localization and Recognition

In this technical roadmap, the basic idea is to use deep neural networks to train complex scene data to obtain a deep learning model FLsM, which predicts the approximate position and scene type of the captured photo based on the image. The schematic diagram is shown in Figure 5. When it needs to achieve fast communications between two arbitrary global points, the satellites in the air platform are used for forwarding communications. The solutions provide differentiated services for the ground user according to the quality, content, and priority. In Figure 5, image and text information fusion processing positioning can be seen.

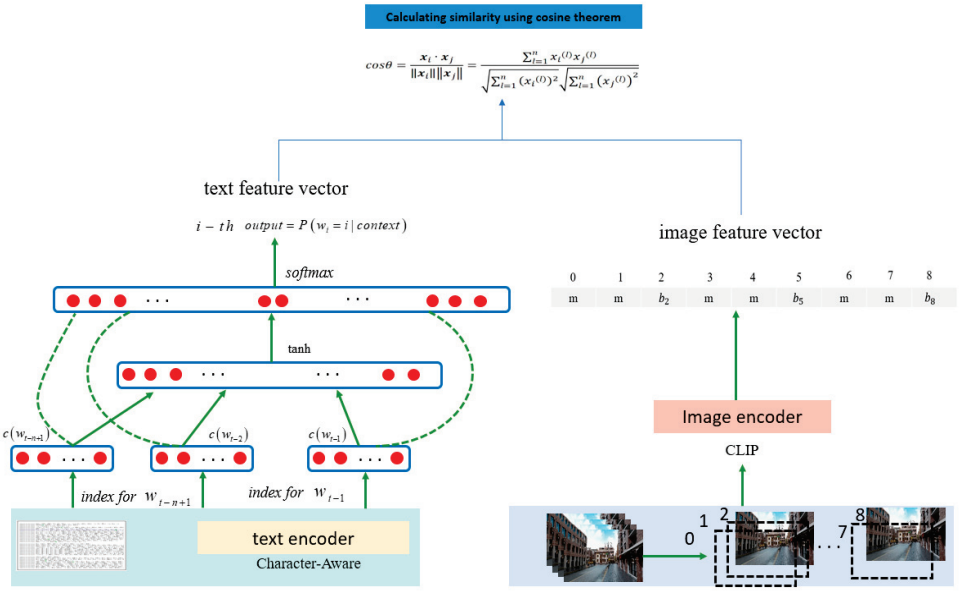


Figure 5. Image and text information fusion processing positioning.

In this technical roadmap, the basic idea is to use deep neural networks to train complex scene data to obtain a deep learning model, FLsM, which predicts the approximate position and scene type of the captured photo, based on the image in Figure 6. According to actual needs, images with GNSS labels are trained on a large amount of data to complete a set of deep learning methods for image localization. The geographic localization problem is transformed into a classification problem; by quantifying all image data with GNSS labels into a fixed number of classes, the GNSS labels are converted into class labels, so that each class represents a physical region in the real world. Then, the classification results are converted into GNSS coordinates of the corresponding region. In this study, in order to obtain a more accurate positioning model, we use multiple models including OpenAI double CLIP model for training [37]. Based on the ability of efficient matching between

images and texts, the large model enhances the generalization ability of the visual image positioning model. More than 400 million pairs of image text data are used for pre-training through a large number of Internet data, which cover a wealth of topics and scenes and provide a wide range of samples for model training. A unique method is used in the training process of the visual image positioning model. First, a batch_size image text pair is selected, and then the image is encoded using Image Encoder and the text is encoded using Text Encoder. Next, the cosine similarity between the encoded image and the text vector is calculated to verify the matching between the image and the text. Thanks to its powerful pre-training ability and effective matching verification method, it can be seen through experiments that the positioning accuracy of the model in multiple scenes has reached the current best performance (SOTA). Based on the CLIP model, the visual image positioning model consists of two parts—a visual encoder and a text encoder. The visual encoder is the part used to process the image, which converts the input image into a vector representation of fixed length. The visual encoder can choose to use either the CNN-based ResNet or the Transformer-based ViT. The text encoder is the part used to process text, which converts the input natural language text sequence into a fixed-length vector representation. The text encoder uses the Transformer model. Both encoders are trained to map the input information into the same embedding space and make similar images and texts closer in the embedding space. Model parameters—in different versions of the CLIP model, the number of parameters is different. To ensure transparency and reproducibility in our study, we provide a summary of the datasets used in our experiments in Table 1. This summary includes key information such as the dataset names, their respective sources, and brief descriptions.

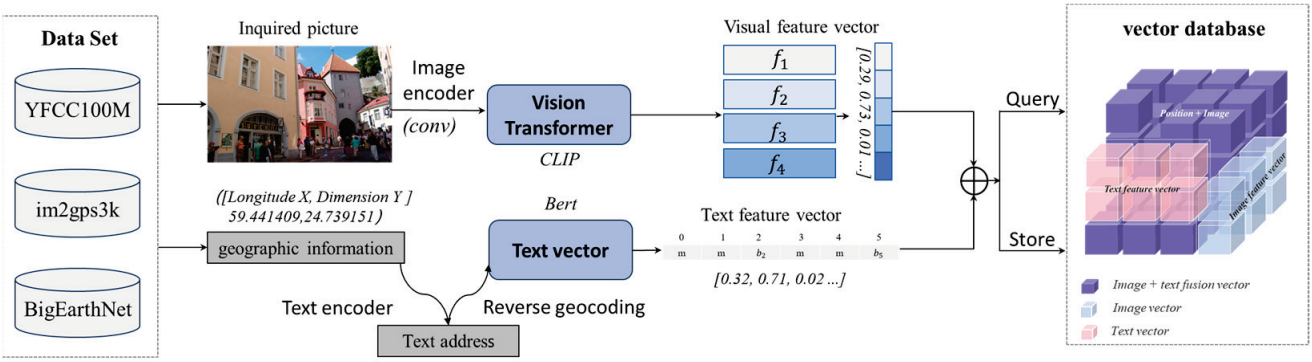


Figure 6. Overall framework of the FLsM structure, integrating image and text large models.

Table 1. Summary of datasets.

Dataset Name	Source	Description
Google Landmarks	cvdfoundation/ google-landmark	5 million landmark-labeled images.
BigEarthNet	Technische Universität Berlin	Contains 590,326 image pairs from Sentinel-1 and Sentinel-2.
Im2GPS3k	TIBHannover/GeoEstimation	Comprises 3000 geotagged images that span a variety of scenes and locations worldwide.
GeoYFCC	abhimanyudubey/GeoYFCC	Comprises a total of 1,147,059 images from 1261 categories across 62 countries.

4. Results

This section delves into investigating the influence of larger models on accuracy through a multi-model and multi-sample approach, using the Google Landmarks Dataset. Specifically, two CLIP models, StreetCLIP (with 420 million parameters) and MetaClip (with 980 million parameters) [38,39], are employed. The experiment adopts a traditional hierarchical search method to facilitate CLIP in deducing the geographical locations of the images. By juxtaposing the performance of these models, particularly highlighting the substantial difference in parameter size between StreetCLIP and MetaClip, valuable insights into the impact of model size on accuracy can be gleaned. The methodology unfolds as follows:

- Step 1: Reverse geocode the latitude and longitude coordinates of the image to obtain textual geographical location information, including country, first-level administrative region, second-level administrative region, address, and detailed address.
- Step 2: Employ the model to predict the country of the image and compare it with the country information obtained from the textual data.
- Step 3: Utilize the model to predict the first-level administrative region of the image and compare it with the corresponding information from the textual data.

  ...

- Continue this iterative process until the detailed address is determined. Then, juxtapose it with the actual region of the image to derive accuracy metrics. The experimental findings are summarized in Table 2 below.

Table 2. Experimental results of the StreetCLIP and MetaClip models.

Model	Country	First-Level Administrative Region	Second-Level Administrative Region	Address	Detailed Address
StreetCLIP	**20.65%**	**6.02%**	1.79%	0.71%	0.59%
MetaCLIP	20.45%	5.99%	**1.93%**	**0.95%**	**0.89%**

Note: Bold green indicates the part with the highest accuracy.

Through experiments, we can see that the positioning accuracy is extremely low and replacing a larger model will not significantly improve the positioning accuracy. Based on this, we change the dataset and select im2gps3k, GeoYFCC, and BigEarthNet for experimental verification [40].

The Im2GPS3k dataset is a subset of the original Im2GPS dataset, which is used for testing in the field of photo geographic positioning estimation. This dataset is an important part of the estimation benchmark of photo geographic location. The purpose of using this dataset is to determine the exact latitude and longitude of the photo shooting place, which is a challenging but widely applicable task in the field of computer vision. There are about 3000 pictures and the dataset distribution, as shown in the figure, of GeoYFCC is a geographical subset of YFCC, which ensures that each country has 20,000–30,000 pictures, so the geographical distribution is more uniform. It contains about 1 million pictures and the dataset is distributed as shown in the figure. BigEarthNet is a large-scale remote sensing dataset based on Sentinel-2 satellite images, which contains 5.9 million image blocks in Europe, each with a size of 120 $\times$ 120 pixels, with 13 spectral bands covering 43 land cover/use types. The dataset distribution is shown in Figure 7.

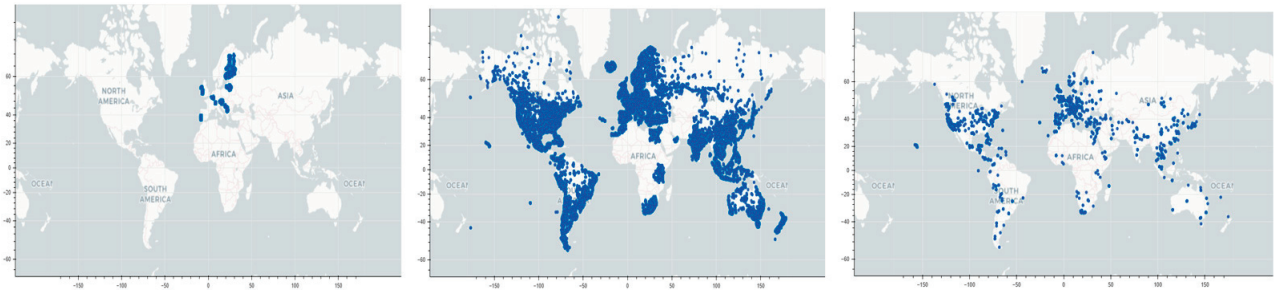


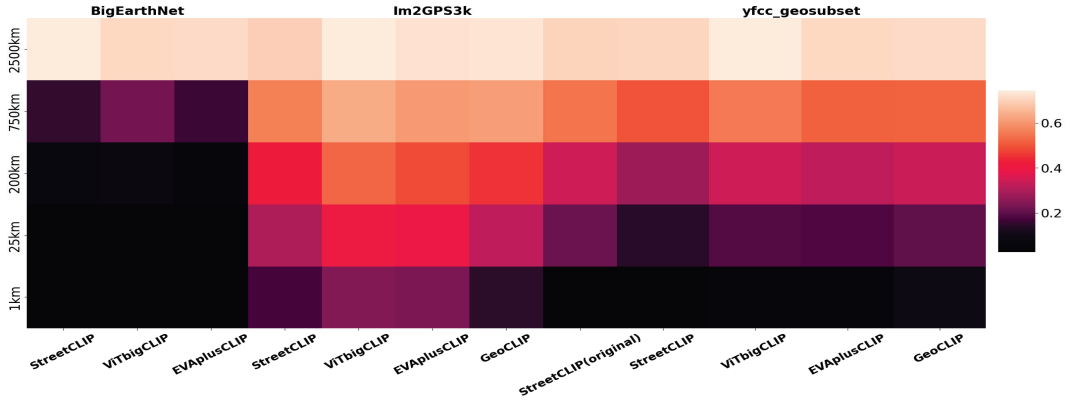
Figure 7. Schematic diagram of global distribution of image dataset.

In the small-scale space under the scene of street and living environment, we studied and performed experiments on two datasets. Firstly, based on the im2gps3k dataset, this study carried out a set of comparative experiments. The traditional hierarchical search is used to traverse the positioning through hierarchical query. First, start the query from a larger area, such as the country, then narrow down the scope one by one, such as the first-level administrative region and the second-level administrative region, and finally obtain the detailed address. The contrast experiment is carried out using a brand-new similarity calculation method. This experiment utilized Milvus as the vector database framework. The specific experimental process is as follows. Initially, each image in the dataset was processed. Every image is associated with a unique image ID and latitude–longitude information. The latitude–longitude coordinates were converted into textual location information using reverse geocoding. Simultaneously, the textual location information was encoded using a model to generate textual feature vectors. These components were abstracted into Milvus entities and stored in the database. These steps were repeated for each image in the dataset. Subsequently, the dataset was traversed again. The model was used to extract image feature vectors from each picture. Then, the database was queried to find the Milvus entity corresponding to the textual feature vector with the highest cosine similarity to the image feature vector. The latitude–longitude coordinates of the image and the corresponding entity were compared to calculate the predicted distance error. Finally, the errors were categorized into different scales (e.g., errors less than 1 km were categorized as within 1 km, errors less than 25 km were categorized as within 25 km, where 25 km includes 1 km). This process aims to establish associations between images and geographical location information. Through the efficient vector search capabilities of the Milvus database, the positioning accuracy of the models was evaluated across different scales. The comparison between the new method using StreetCLIP and the traditional method using StreetCLIP shows that the comparison between the new method and the traditional method is obviously improved on a smaller scale, but there is little difference on a larger scale. Then, new methods are used to test the performance of other models in image positioning, including ViTbigCLIP (GeoDE dataset performs well) and EVAplusCLIP (model parameters are large, reaching 5 billion). The performance can be further improved after replacing other better models (e.g., ViTbigCLIP performs better on GeoDE, and EVAplusCLIP model parameters are larger). The following experimental results are shown in Table 3. Thermal maps of positioning accuracy are shown in Figure 8a. On the GeoYFCC dataset, we use the same method to test StreetCLIP, ViTbigCLIP, EVAplusCLIP, and GeoCLIP (because CeoCLIP has no text encoder, so we use the traditional hierarchical search method for this model). The experimental results are shown in Figure 8b,c. Based on remote sensing images in large-scale space, the positioning accuracy of the model is tested on the BigEarthNet dataset, as shown in Figure 8d. Through experiments, it can be found that the overall accuracy of positioning based on remote sensing images is low, and it only improves slightly at the scale of 750 km, but the partial area of this dataset is mostly around 2500 km, which leads to the soaring accuracy of the last 2500 km, so it is more effective in large-scale space.

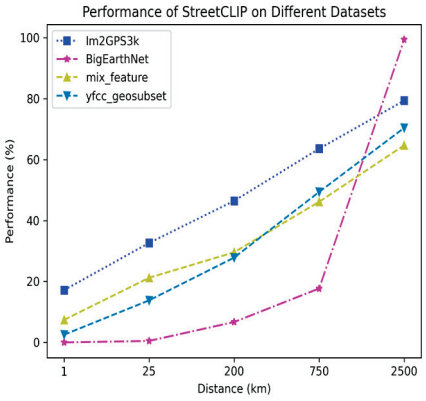
Table 3. Experimental results under different models.

Dataset	Model	1 km_Accuracy	25 km_Accuracy	200 km_Accuracy	750 km_Accuracy	2500 km_Accuracy
BigEarthNet	StreetCLIP	**3.38796 × 10^{-06}**	**0.004997239**	0.067010432	0.177152963	**0.99427435**
BigEarthNet	ViTbigCLIP	2.20217 × 10^{-05}	0.004595766	**0.071401226**	**0.274355187**	0.950119087
BigEarthNet	EVAplusCLIP	1.18579 × 10^{-05}	0.004094348	0.042068281	0.18899896	0.956994949
Im2GPS3k	StreetCLIP	0.171838505	0.326659993	0.464798131	0.635969303	0.794127461
Im2GPS3k	ViTbigCLIP	**0.256256256**	**0.450784117**	**0.589255923**	**0.724724725**	**0.851851852**
Im2GPS3k	EVAplusCLIP	0.249249249	0.431097764	0.544210878	0.688688689	0.829496163
mix_feature	StreetCLIP	**0.074074074**	**0.212545879**	**0.297297297**	**0.464130797**	**0.650650651**
mix_feature	ViTbigCLIP	0.016016016	0.082749416	0.122455789	0.23023023	0.448114781
mix_feature	EVAplusCLIP	0.048381715	0.125792459	0.18685352	0.294627961	0.515181849

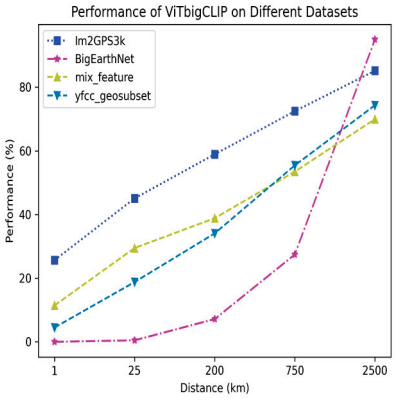
Note: The accuracy values represent the proportion of correctly identified geographical locations. The spatial scales are in kilometers. Bold green indicates the part with the highest accuracy.



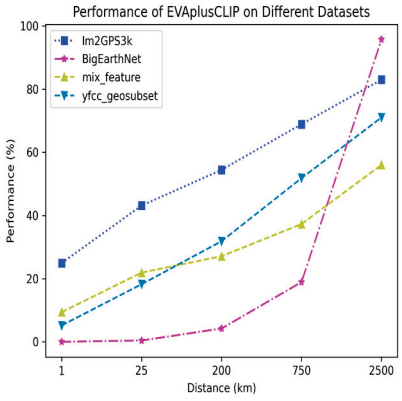
(**a**) Thermal maps of positioning accuracy under different models



(**b**) StreetCLIP



(**c**) ViTbigCLIP



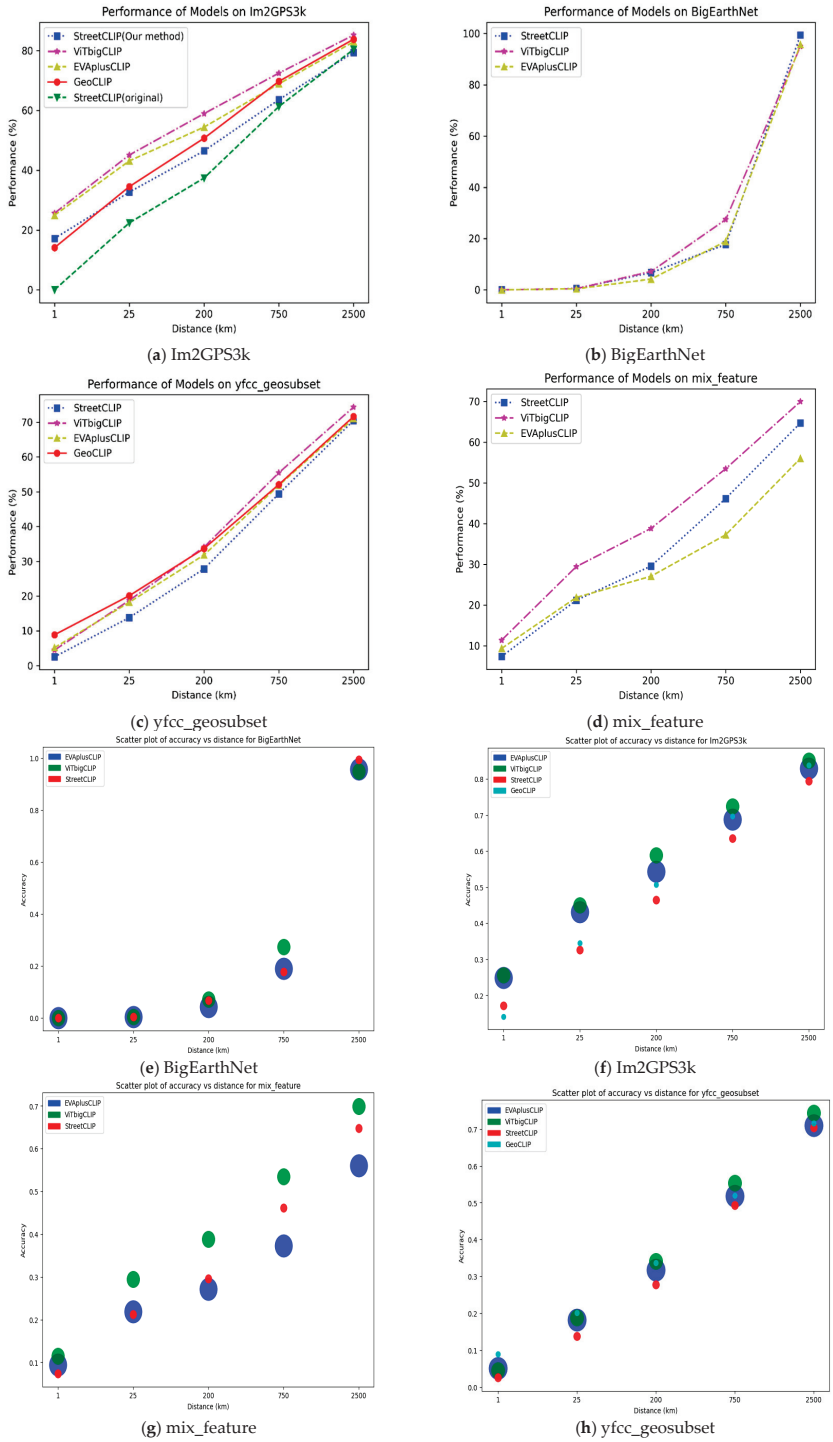
(**d**) EVAplusCLIP

Experimental results of positioning progress of different datasets under different models.

Figure 8. Experimental results under different models.

Examine the four charts in Figure 9a–d to see how each model performs under various distance settings and datasets. In most cases, the enhanced version of StreetCLIP (designated as "Our method") has demonstrated superior performance, particularly in large-scale spaces where it operates more flawlessly. On some datasets, the StreetCLIP model outperforms the VitbigCLIP and EVAplusCLIP models; however, the enhanced version of StreetCLIP exhibits a more consistent performance growth in a number of areas. In every scenario, the StreetCLIP original version displayed the slowest performance growth. Figure 9e–h shows the accuracy performance of many models under various distance parameters on four distinct datasets. The variation in the model's performance is repre-

sented by the scatter's size. "Our method" (StreetCLIP) generally shows rather large scatter points in all datasets, especially when covering big distances (750 km and 2500 km), which suggests high accuracy. In comparison to other models, "Our method" exhibits a notable improvement in accuracy, particularly on the lm2GPS3k and yfcc_geosubset datasets.



(**a**) Im2GPS3k (**b**) BigEarthNet

(**c**) yfcc_geosubset (**d**) mix_feature

(**e**) BigEarthNet (**f**) Im2GPS3k

(**g**) mix_feature (**h**) yfcc_geosubset

Figure 9. Relationship between positioning accuracy of different datasets under different models.

5. Discussion

To investigate visual blur positioning using large models, we utilized conventional street/environmental image and remote sensing image datasets as elastic scale space samples. Experiments were conducted across various large models to analyze their performance. The results indicate that models with larger parameters, such as EVAplusCLIP, exhibit enhanced positioning accuracy on elastic scale space sample datasets, with more stable outcomes aligning with the requirements of visual blur in elastic scale space positioning. The key advantage of EVAplusCLIP lies in its larger model parameters, enabling better data complexity capture and representation, thereby enhancing model generalization. Notably, the output vector dimension of the model significantly impacts its performance, as evidenced by the ViTbigCLIP model having the highest positioning accuracy score on the GeoDE dataset, due to its larger vector dimensions providing more information for improved prediction accuracy. However, optimizing model parameters and vector dimensions alone may not suffice to meet all requirements. To further enhance model accuracy, alternative improvement methods should be considered, such as introducing a vector database to optimize vector retrieval and utilization for improved model accuracy.

6. Conclusions

In this research, we introduce a fuzzy positioning approach for images based on large models in elastic scale space. Our comparative experiments demonstrate that the EVAplusCLIP model achieves a higher positioning accuracy and can effectively serve the image positioning function across various scale spaces. This work represents an exploratory research endeavor with several areas open for future improvement. Potential research directions include optimizing model stability through further experiments with increased model parameters, enhancing model performance on specific datasets by expanding the output vector dimension and training on more relevant data, and exploring additional improvement methods such as vector databases to enhance model accuracy. These paths present critical avenues for our ongoing in-depth investigation and optimization of this research. Through conducting additional experiments and exploring further improvement methods, we can enhance the stability and accuracy of our image positioning approach. Our research has produced promising results, indicating the potential for further advances in model stability through increased parameter experiments. Furthermore, expanding output vector dimensions and training on more relevant data offer exciting opportunities for enhancing model performance across specific datasets. Incorporating vector databases as an improvement method also introduces new possibilities for optimizing positioning accuracy. These areas provide essential directions for our ongoing in-depth investigation and advancement of this research.

Author Contributions: Conceptualization, W.C.; Methodology, W.C.; Formal analysis, Y.W.; Resources, Y.L.; Writing—original draft, W.C.; Supervision, L.M. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [China's Ministry of Science and Technology National Key R&D Program Beidou xing Energy] grant number [E33514060C].

Data Availability Statement: The data presented in this study are available in this article.

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

References

1. Li, X.; Wang, B.; Li, X.; Huang, J.; Lyu, H.; Han, X. Principle and performance of multi-frequency and multi-GNSS PPP-RTK. *Satell. Navig.* **2022**, *3*, 7. [CrossRef]
2. Al Hage, J.; Najjar, M.E.B.E. Improved Outdoor Localization Based on Weighted Kullback-Leibler Divergence for Measurements Diagnosis. *IEEE Intell. Transp. Syst. Mag.* **2018**, *12*, 41–56. [CrossRef]
3. Li, M.-H.; Jiang, P.; Yu, D.-J.; Sun, J.-H. Position and attitude determination by integrated GPS/SINS/TS for feed support system of FAST. *Res. Astron. Astrophys.* **2020**, *20*, 140. [CrossRef]

4. Shi, Q.; Wu, J.; Lin, Z.; Qin, N. Learning a Robust Hybrid Descriptor for Robot Visual Localization. *J. Robot.* **2022**, *2022*, 9354909. [CrossRef]
5. Wang, Z.; Cheng, X. Adaptive optimization online IMU self-calibration method for visual-inertial navigation systems. *Measurement* **2021**, *180*, 109478. [CrossRef]
6. Cao, M.; Tang, F.; Ji, P.; Ma, F. Improved Real-Time Semantic Segmentation Network Model for Crop Vision Navigation Line Detection. *Front. Plant Sci.* **2022**, *13*, 898131. [CrossRef]
7. Critchley-Marrows, J.J.; Wu, X.; Cairns, I.H. An architecture for a visual-based PNT alternative. *Acta Astronaut.* **2023**, *210*, 601–609. [CrossRef]
8. Bellamy, B.R. The Robotic Imaginary: The Human and the Price of Dehumanized Labor by Jennifer Rhee. *Sci. Fict. Stud.* **2019**, *46*, 655–657. [CrossRef]
9. Cass, S. Ayanna Howard: Robot wrangler. *IEEE Spectr.* **2005**, *42*, 21–22. [CrossRef]
10. Carpenter, K.; Wiltsie, N.; Parness, A. Rotary Microspine Rough Surface Mobility. *IEEE/ASME Trans. Mechatron.* **2015**, *21*, 2378–2390. [CrossRef]
11. Tang, C.; Ma, W.; Li, B.; Jin, M.; Chen, H. Cephalopod-Inspired Swimming Robot Using Dielectric Elastomer Synthetic Jet Actuator. *Adv. Eng. Mater.* **2019**, *22*, 1901130. [CrossRef]
12. Ning, X.; Liu, L. A Two-Mode INS/CNS Navigation Method for Lunar Rovers. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 2170–2179. [CrossRef]
13. Ning, X.; Fang, J. A new autonomous celestial navigation method for the lunar rover. *Robot. Auton. Syst.* **2009**, *57*, 48–54. [CrossRef]
14. Wang, W.-R.; Ren, X.; Wang, F.-F.; Liu, J.-J.; Li, C.-L. Terrain reconstruction from Chang'e-3 PCAM images. *Res. Astron. Astrophys.* **2015**, *15*, 1057–1067. [CrossRef]
15. Sutoh, M.; Wakabayashi, S.; Hoshino, T. Influence of atmosphere on lunar rover performance analysis based on soil parameter identification. *J. Terramech.* **2017**, *74*, 13–24. [CrossRef]
16. Choi, I.-S.; Ha, J.-E. Simple method for calibrating omnidirectional stereo with multiple cameras. *Opt. Eng.* **2011**, *50*, 43608. [CrossRef]
17. Gamarra, D.F.T.; Pinpin, L.K.; Laschi, C.; Dario, P. Forward Models Applied in Visual Servoing for a Reaching Task in the iCub Humanoid Robot. *Appl. Bionics Biomech.* **2009**, *6*, 345–354. [CrossRef]
18. Zhang, M.; Cui, J.; Zhang, F.; Yang, N.; Li, Y.; Li, F.; Deng, Z. Research on evaluation method of stereo vision measurement system based on parameter-driven. *Optik* **2021**, *245*, 167737. [CrossRef]
19. Huang, W.; Fajen, B.R.; Fink, J.; Warren, W.H. Visual navigation and obstacle avoidance using a steering potential function. *Robot. Auton. Syst.* **2006**, *54*, 288–299. [CrossRef]
20. Bulanon, D.; Burks, T.; Alchanatis, V. Image fusion of visible and thermal images for fruit detection. *Biosyst. Eng.* **2009**, *103*, 12–22. [CrossRef]
21. Kuo, B.-W.; Chang, H.-H.; Chen, Y.-C.; Huang, S.-Y. A Light-and-Fast SLAM Algorithm for Robots in Indoor Environments Using Line Segment Map. *J. Robot.* **2011**, *2011*, 257852. [CrossRef]
22. Lv, K.; Zhang, Y.; Yu, Y.; Wang, Z.; Min, J. SIIS-SLAM: A Vision SLAM Based on Sequential Image Instance Segmentation. *IEEE Access* **2022**, *11*, 17430–17440. [CrossRef]
23. Zhao, X.; Wang, T.; Li, Y.; Zhang, B.; Liu, K.; Liu, D.; Wang, C.; Snoussi, H. Target-Driven Visual Navigation by Using Causal Intervention. *IEEE Trans. Intell. Veh.* **2023**, *9*, 1294–1304. [CrossRef]
24. Li, J.; Yin, J.; Deng, L. A robot vision navigation method using deep learning in edge computing environment. *EURASIP J. Adv. Signal Process.* **2021**, *2021*, 22. [CrossRef]
25. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Learning to Prompt for Vision-Language Models. *Int. J. Comput. Vis.* **2022**, *130*, 2337–2348. [CrossRef]
26. Xing, Y.; Wu, Q.; Cheng, D.; Zhang, S.; Liang, G.; Wang, P.; Zhang, Y. Dual Modality Prompt Tuning for Vision-Language Pre-Trained Model. *IEEE Trans. Multimed.* **2023**, *26*, 2056–2068. [CrossRef]
27. Sun, B.; Liu, G.; Yuan, Y. F3-Net: Multiview Scene Matching for Drone-Based Geo-Localization. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 3278257. [CrossRef]
28. Vicente Vivanco, C.; Nayak, G.K.; Shah, M. GeoCLIP: Clip-Inspired Alignment between Locations and Images for Effective Worldwide Geo-localization. *arXiv* **2023**, arXiv:2309.16020.
29. Gao, H.; Zhu, M.; Wang, X.; Li, C.; Xu, S. Lightweight Spatial-Spectral Network Based on 3D-2D Multi-Group Feature Extraction Module for Hyperspectral Image Classification. *Int. J. Remote Sens.* **2023**, *44*, 3607–3634. [CrossRef]
30. Cheng, P.; Cheng, Y.; Wang, X.; Wu, S.; Xu, Y. Realization of an Optimal Dynamic Geodetic Reference Frame in China: Methodology and Applications. *Engineering* **2020**, *6*, 879–897. [CrossRef]
31. Thomee, B.; Shamma, D.A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; Li, L.-J. Yfcc100m: The new data in multimedia research. *arXiv* **2016**, arXiv:1503.01817. [CrossRef]
32. Alsubai, S.; Dutta, A.K.; Alkhayyat, A.H.; Jaber, M.M.; Abbas, A.H.; Kumar, A. Hybrid deep learning with improved Salp swarm optimization based multi-class grape disease classification model. *Comput. Electr. Eng.* **2023**, *108*, 108733. [CrossRef]
33. Anguelov, D.; Dulong, C.; Filip, D.; Frueh, C.; Lafon, S.; Lyon, R.; Ogale, A.; Vincent, L.; Weaver, J. Google Street View: Capturing the World at Street Level. *Computer* **2010**, *43*, 32–38. [CrossRef]

34. Steven, B.; Ayton, A. Text-to-Image Synthesis with Self-supervision via Contrastive Language-Image Pre-Training (CLIP). Available online: https://www.researchgate.net/publication/369299175_Text-to-Image_Synthesis_with_Self-supervision_via_Contrastive_Language-Image_Pre-training_CLIP (accessed on 13 May 2024).
35. Vo, N.; Jacobs, N.; Hays, J. Revisiting im2gps in the deep learning era. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
36. Sumbul, G.; Wall, A.; Kreuziger, T.; Marcelino, F.; Costa, H.; Benevides, P.; Caetano, M.; Demir, B.; Markl, V. BigEarthNet-MM: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 174–180. [CrossRef]
37. Roumeliotis, K.I.; Tselikas, N.D. ChatGPT and Open-AI Models: A Preliminary Review. *Futur. Internet* **2023**, *15*, 192. [CrossRef]
38. Haas, L.; Silas, A.; Michal, S. Learning generalized zero-shot learners for open-domain image geolocalization. *arXiv* **2023**, arXiv:2302.00275.
39. Parashar, S.; Lin, Z.; Liu, T.; Dong, X.; Li, Y.; Ramanan, D.; Caverlee, J.; Kong, S. The Neglected Tails of Vision-Language Models. *arXiv* **2024**, arXiv:2401.12425.
40. Dubey, A.; Ramanathan, V.; Pentland, A.; Mahajan, D. Adaptive methods for real-world domain generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021.

electronics

MDPI

Article

IPCB: Intelligent Pseudolite Constellation Based on High-Altitude Balloons

Yi Qu [1,2,*], Sheng Wang [1,2], Tianshi Pan [1] and Hui Feng [1]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: quyi@aircas.ac.cn

Abstract: IPCBs (Intelligent Pseudolite Constellations based on high-altitude balloons) are a novel type of air-based pseudolite application with many advantages. Compared with ground-based pseudolites and traditional air-based pseudolites, IPCBs have a wider coverage and a lower energy requirement. Compared with LEO satellite constellations, IPCBs have a stronger signal, a lower cost, and a shorter deployment period. These merits give promising potential to IPCBs. In IPCB applications, one of the key factors is geometry configuration, which is deeply influenced by the balloon's unique features. The basic idea of this paper is to pursue a strategy to improve IPCB geometry performance by using diverse winds at different altitudes and balloons' capability of altering flight altitude intelligently. Starting with a brief introduction to IPCBs, this paper defines an indicator to assess IPCB geometry performance, an approach to adjust IPCB geometry configuration and an IPCB geometry configuration planning algorithm. Next, a series of simulations are implemented with an IPCB composed of six pseudolites in winds with/without a quasi-zero wind layer. Some IPCB geometry configurations are analyzed, and their geometry performances are compared. Simulation results show the effectiveness of the proposed algorithm and the influence of the quasi-zero wind layer on IPCB performance.

Keywords: intelligent pseudolite; constellation; high altitude balloon

1. Introduction

Pseudolites are transmitters that can emit navigation signals to improve GNSS performance or to provide navigation service independently [1–7]. An intelligent pseudolite based on high-altitude balloons (IPBs) is a type of novel air-based pseudolite that utilizes a high-altitude balloon as a platform. An intelligent pseudolite constellation based on high-altitude balloons (IPCBs) is composed of multiple IPBs, which can provide emergency positioning service and regional positioning service independently.

As an excellent solution for regional positioning, ICPBs have many advantages. Compared with ground-based pseudolites and traditional air-based pseudolites, IPCBs have a wider coverage and can serve more users because their flight altitude can reach tens of kilometers [8,9]. Furthermore, IPCBs have a low energy requirement since they can accomplish its flight primarily relying on buoyancy and wind rather than oil or electricity [10,11]. Compared with LEO satellite constellations, IPCBs have a stronger signal, a lower cost, and a shorter deployment period [12,13]. In addition, the continuous residence duration of an IPB over a service area is longer than that of an LEO satellite, which can reduce navigation signal lock-lose and cycle slip caused by satellite switching [14–16]. Also, the IPCB operation and maintenance burden is less than that of an LEO satellite constellation because an LEO satellite constellation usually comprises a large number of satellites [17,18]. These merits give promising potential to IPCB applications.

In the application of IPCBs, geometry configuration plays a critical role since it affects IPCB service performance significantly [19–22]. However, the problem becomes very

complicated because of the unique dynamic features of IPBs. Most traditional research studies about pseudolite geometry configuration are designed for static pseudowires, which are not suitable for IPCBs [23–27].

This paper centers on the problem of IPCB geometry configuration and proposes a planning algorithm that emphasizes utilizing different winds smartly. The proposed algorithm can fit the dynamic flight of IPCBs adaptively and is easy to implement. Moreover, it can achieve performance improvements by controlling IPB valves and fans only, without extra hardware cost.

The rest of this paper is organized as follows. An overview of IPCBs is described in Section 2, a performance indicator of IPCB geometry configuration is defined in Section 3, an IPCB geometry configuration adjustment approach is designed in Section 4, a series of constraints are discussed in Section 5, a planning algorithm based on whale optimization algorithm (WOA) is proposed in Section 6, simulations and discussions are presented in Section 7, and conclusions are stated in Section 8.

2. Overview of IPCB

An IPB is illustrated in Figure 1. It utilizes a high-altitude balloon as a platform, which is composed of balloon, cable, parachute, gondola, balloon controller, payloads, and other attachments, as shown in Figure 1a. The balloon controller, payloads, and other attachments are installed in the gondola. Furthermore, the balloon consists of a main helium bag and an air ballonet, which are separated by a membrane, as Figure 1b illustrates [28–30]. The main helium bag is filled with helium to provide buoyancy to the IPB, and the air ballonet is filled with air to adjust the IPB mass. Fans and valves are installed on the bottom of the balloon, which can pump air into or release air from the air ballonet. The balloon controller can perceive and adjust IPB flight status intelligently. In particular, the balloon controller can manipulate the fans and valves flexibly, enabling the IPB to adjust its mass and flight altitude in a range.

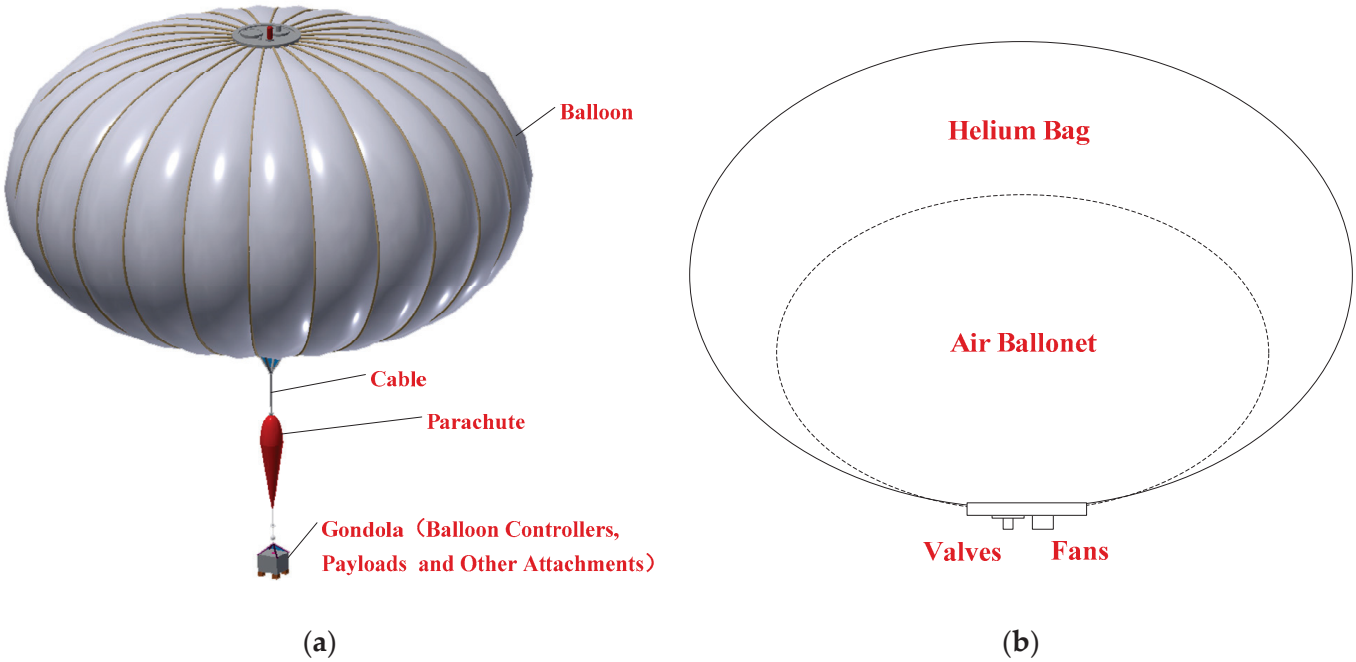


Figure 1. Structure illustration of IPB studied in this paper: (**a**) structure overview; (**b**) detailed structure of the balloon.

An IPCB is illustrated in Figure 2. It consists of multiple IPBs and can form coverage over a service area. When payloads in the IPBs normally send out navigation signals, the IPCB can provide positioning service for the area independently. During the process of

IPCB service, its geometry configuration is always changing with the wind, and the changes are highly nonlinear, which brings difficulty to IPCB geometry configuration planning.

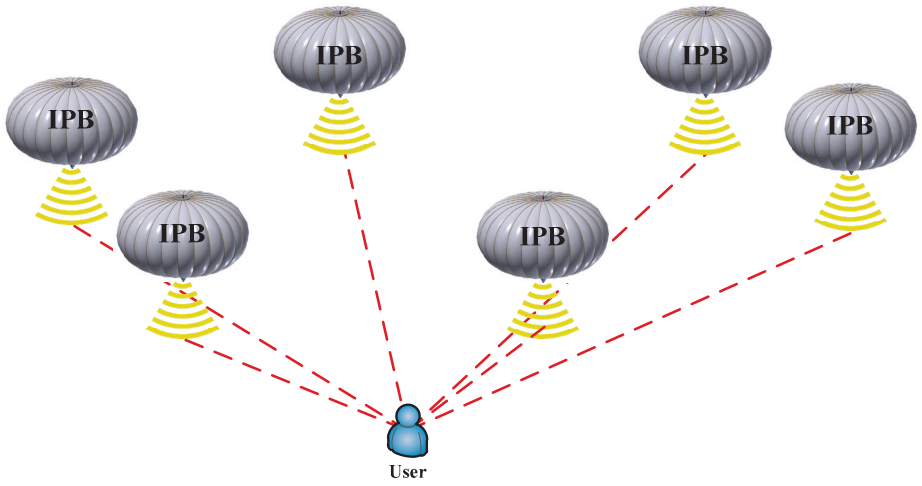


Figure 2. Illustration of IPCB in regional positioning.

3. Performance Indicator of IPCB Geometry Configuration

Assuming that an IPCB composed of n_p IPBs is deployed above the service area initially and that n_u users are selected as samples to assess IPCB geometry performance. Then, the pseudo-range equation from the j-th IPB to the i-th user at time t can be expressed by Equation (1) [31–33].

$$\rho(i,j,t) = \sqrt{(x(j,t) - x_u(i,t))^2 + (y(j,t) - y_u(i,t))^2 + (z(j,t) - z_u(i,t))^2} + ct_u(i,t) \quad (1)$$

In Equation (1), $\rho(i,j,t)$ represents the pseudo-range from the j-th IPB to the i-th user at time t; $(x(j,t), y(j,t), z(j,t))$ represents the position of the j-th IPB at time t; $(x_u(i,t), y_u(i,t), z_u(i,t))$ represents the position of the i-th user at time t; c represents the speed of light; $t_u(i,t)$ represents the clock difference of the i-th user at time t.

Equation (1) can be rewritten as Equation (2) after the first-order Taylor expansion.

$$\Delta\rho(i,j,t) = a_x(i,j,t)\Delta x_u(i,t) + a_y(i,j,t)\Delta y_u(i,t) + a_z(i,j,t)\Delta z_u(i,t) - c\Delta t_u(i,t) \quad (2)$$

In Equation (2), 'Δ' represents the difference between the Taylor expansion point and its neighborhood, $(a_x(i,j,t), a_y(i,j,t), a_z(i,j,t))$ represents the direction cosine from the i-th user to the j-th IPB at time t, which can be calculated by Equation (3).

$$\begin{aligned} a_x(i,j,t) &= \frac{x(j,t) - x_u(i,t)}{\sqrt{(x(j,t) - x_u(i,t))^2 + (y(j,t) - y_u(i,t))^2 + (z(j,t) - z_u(i,t))^2}} \\ a_y(i,j,t) &= \frac{y(j,t) - y_u(i,t)}{\sqrt{(x(j,t) - x_u(i,t))^2 + (y(j,t) - y_u(i,t))^2 + (z(j,t) - z_u(i,t))^2}} \\ a_z(i,j,t) &= \frac{z(j,t) - z_u(i,t)}{\sqrt{(x(j,t) - x_u(i,t))^2 + (y(j,t) - y_u(i,t))^2 + (z(j,t) - z_u(i,t))^2}} \end{aligned} \quad (3)$$

Equation (2) can be expanded from the j-th IPB to all the IPBs in the IPCB, which can be simplified as

$$\boldsymbol{\Delta\rho}(i,t) = \boldsymbol{H}(i,t)\boldsymbol{\Delta x}(i,t) \quad (4)$$

In Equation (4), $\boldsymbol{\Delta\rho}(i,t)$ is a vector representing pseudo-range measurement error, $\boldsymbol{\Delta x}(i,t)$ is a vector representing positioning error, and $\boldsymbol{H}(i,t)$ is an observation matrix.

Equation (4) can be rewritten as Equation (5) by the least square method.

$$\boldsymbol{\Delta x}(i,t) = (\boldsymbol{H}(i,t)^T\boldsymbol{H}(i,t))^{-1}\boldsymbol{H}(i,t)^T\boldsymbol{\Delta\rho}(i,t) \quad (5)$$

If the pseudo-range noises of different IPBs are linearly independent, with an average of 0 and a variance of σ^2, then the covariance of $\boldsymbol{\Delta x}(i,t)$ can be expressed by Equation (6) [33].

$$\mathrm{cov}(\boldsymbol{\Delta x}(i,t)) = \sigma^2(\boldsymbol{H}(i,t)^{\mathrm{T}}\boldsymbol{H}(i,t))^{-1} \tag{6}$$

From Equation (6), it can be concluded that $(\boldsymbol{H}(i,t)^{\mathrm{T}}\boldsymbol{H}(i,t))^{-1}$ reveals the magnification from a user pseudo-range measurement error to its positioning error. Given the same user pseudo-range measurement error, the smaller $(\boldsymbol{H}(i,t)^{\mathrm{T}}\boldsymbol{H}(i,t))^{-1}$ is, the smaller the positioning error is. Consequently, the square root of the trace of $(\boldsymbol{H}(i,t)^{\mathrm{T}}\boldsymbol{H}(i,t))^{-1}$ is usually treated as an important indicator to assess the influence of a constellation geometry configuration on its positioning error, named GDOP (geometric dilution of precision), which can be expressed by Equation (7) [33–36].

$$GDOP(i,t) = \sqrt{tr(\boldsymbol{H}(i,t)^{T}\boldsymbol{H}(i,t))^{-1}} \tag{7}$$

It is obvious that the value of $GDOP(i,t)$ will fluctuate due to IPCBs' dynamic movements. In particular, some IPBs in the constellation may leave approved airspace as time goes on. In such cases, they cannot continue emitting navigation signals (detailed discussion in Section 5.1), which will decrease the number of available IPBs in the constellation. Once the number of available IPBs in the constellation drops below 4, the GDOP of the IPCB is defined as infinity in this paper.

The geometry performance of an IPCB at time t can be assessed by the average GDOP of multiple users distributed in the service area, as described in Equation (8) [37].

$$GDOP(t) = \frac{1}{n_u}\sum_{i=1}^{n_u} GDOP(i,t) \tag{8}$$

It can be concluded that the objective of IPCB geometry configuration planning is to obtain the minimal $GDOP(t)$ for the whole service duration. However, this approach encounters difficulties when dealing with infinite values. To avoid such difficulties, this paper defines the performance indicator as Equation (9).

$$F = \frac{1}{n_t}\sum_{t=1}^{n_t} \frac{1}{GDOP(t)} \tag{9}$$

In Equation (9), n_t represents the expected service duration of the IPCB. The ultimate objective of IPCB geometry configuration planning is to maximize F defined in Equation (9).

4. Adjustment Approaches of IPCB Geometry Configuration

As discussed in Section 2, IPBs have little actuation capability since they are not equipped with propellers. The primary actuation they can implement is to control their valves and fans, which cannot change the IPCB geometry configuration directly. Therefore, this paper adopts an indirect adjustment approach.

In the vertical direction, IPBs can actively change their masses and flight altitudes by switching their valves and fans. Specific adjustments of each IPB can be managed by its balloon controller. In the horizontal direction, IPBs can change their trajectories with the help of local winds at different altitudes [38–40]. By combining the adjustments in the two directions, IPBs can change their flight trajectories, and the IPCB can modify its geometry configuration [9,28,41,42].

Subsequent detailed analyses will be presented based on the vertical adjustment and the horizontal adjustment of individual IPBs, respectively. The following assumptions are made to simplify the analysis.

(1) The flight altitude of an IPB is fully controllable, and its variation depends on a rise rate and a sink rate, represented by v_{rise} and $v_{\mathrm{sink,}}$ respectively.
(2) The horizontal velocity of an IPB is proportional to the local horizontal wind velocity.

(3) Wind velocities vary with altitude but not with horizontal location. The wind velocities and atmosphere environment are steady during IPCB service time.
(4) The influences of balloon volume variation and thermal effect on IPB motions are ignored.

4.1. Vertical Adjustment of an IPB

An IPB usually keeps its flight altitude by maintaining the balance between gravity and buoyancy, as Equation (10) shows:

$$\boldsymbol{B} = \boldsymbol{G} \tag{10}$$

In Equation (10), $\boldsymbol{B}$ represents its buoyancy, and $\boldsymbol{G}$ represents its gravity. Buoyancy $\boldsymbol{B}$ is related to the atmospheric density, volume of the balloon, and gravitation acceleration, as shown in Equation (11).

$$\boldsymbol{B} = \rho_{air}(h)V\boldsymbol{g} \tag{11}$$

In Equation (11), V represents the volume of the balloon, $\boldsymbol{g}$ represents gravitation acceleration, h represents the flight altitude of the IPB, and $\rho_{air}(h)$ represents the air density at altitude h. The air density is not a constant, and it varies in a wide range. According to the standard atmosphere model (U.S. Standard Atmosphere, 1976), its change with altitude is illustrated in Figure 3.

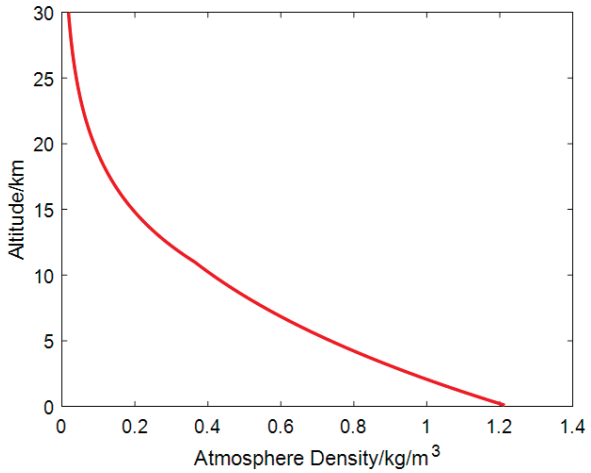


Figure 3. Atmosphere density from 0 km to 30 km.

Gravity $\boldsymbol{G}$ is related to helium mass, air mass, and the masses of others, as shown in Equation (12).

$$\boldsymbol{G} = (m_{He} + m_{air}(h) + m_{other})\boldsymbol{g} \tag{12}$$

In Equation (12), m_{He} represents the helium mass in the main helium bag, $m_{air}(h)$ represents the air mass in the air ballonet at altitude h, m_{other} represents the gross mass of balloon envelop, cable, parachute, gondola, balloon controller, payloads and other attachments.

So if a flight altitude decline is needed, an IPB can pump air into its air ballonet, which will increase its air mass $m_{air}(h)$ and gravity $\boldsymbol{G}$, making $\boldsymbol{G}$ greater than $\boldsymbol{B}$. The change in air mass can be estimated by Equation (13).

$$\Delta m = (\rho_{air}(h_s) - \rho_{air}(h_d))V \tag{13}$$

In Equation (13), h_s and h_d represent the flight altitude before adjustment and after adjustment, respectively. Due to the assumption (1) in this section, the time cost of the flight altitude adjustment Δt can be estimated by Equation (14).

$$h_s - h_d = v_{\sin k}\Delta t \tag{14}$$

In contrast, if a flight altitude ascent is needed, the IPB should release air from its air ballonet, and the time cost can be estimated by Equation (15).

$$h_s - h_d = v_{\text{rise}} \Delta t \tag{15}$$

4.2. Horizontal Adjustment of an IPB

According to assumption (2) in this section, the horizontal velocity of an IPB is proportional to the local horizontal wind velocity [38–40]. Then, the horizontal kinematics of an IPB can be described by Equation (16).

$$\begin{aligned} \dot{\lambda} &= \gamma \cdot w_z(\lambda, \phi, h, t) \\ \dot{\phi} &= \gamma \cdot w_m(\lambda, \phi, h, t) \end{aligned} \tag{16}$$

In Equation (16), λ represents the longitude of IPB position; $\dot{\lambda}$ represents the variation of λ; ϕ represents the latitude of IPB position; $\dot{\phi}$ represents the variation of ϕ; $w_z(\lambda, \phi, h, t)$ represents zonal wind velocity at position (λ, ϕ, h) and at time t; $w_m(\lambda, \phi, h, t)$ represents meridional wind velocity at position (λ, ϕ, h) and at time t; γ represents a drag coefficient of IPB in the horizontal plane.

Since it is assumed that wind velocities vary with altitude but not with horizontal location, and wind velocities are steady in IPCB service time, the relation between altitude and wind velocity is emphasized in this paper. In general, wind velocity increases as altitude increases in the troposphere, reaching a maximum of about 10~15 km. Then, wind velocity decreases, reaching a minimum in the lower portion of the stratosphere at about 18~25 km [43,44]. For simplicity, a seventh-order polynomial is employed to fit the relation between altitude and wind velocity in this paper, as described below [45–47].

$$\begin{aligned} w_m &= c_{m0} + c_{m1} h_{std} + c_{m2} {h_{std}}^2 + \cdots + c_{m7} {h_{std}}^7 \\ w_z &= c_{z0} + c_{z1} h_{std} + c_{z2} {h_{std}}^2 + \cdots + c_{z7} {h_{std}}^7 \end{aligned} \tag{17}$$

In Equation (17), $\boldsymbol{c_m}$ and $\boldsymbol{c_z}$ represent meridional wind coefficients and zonal wind coefficients, respectively. h_{std} represents normalized altitude, which can be calculated by Equation (18).

$$h_{std} = (h - \mu_d)/\sigma_d \tag{18}$$

In Equation (18), μ_d and σ_d are both normalized parameters. Figure 4 illustrates wind fittings for a specific area in March, June, September, and December.

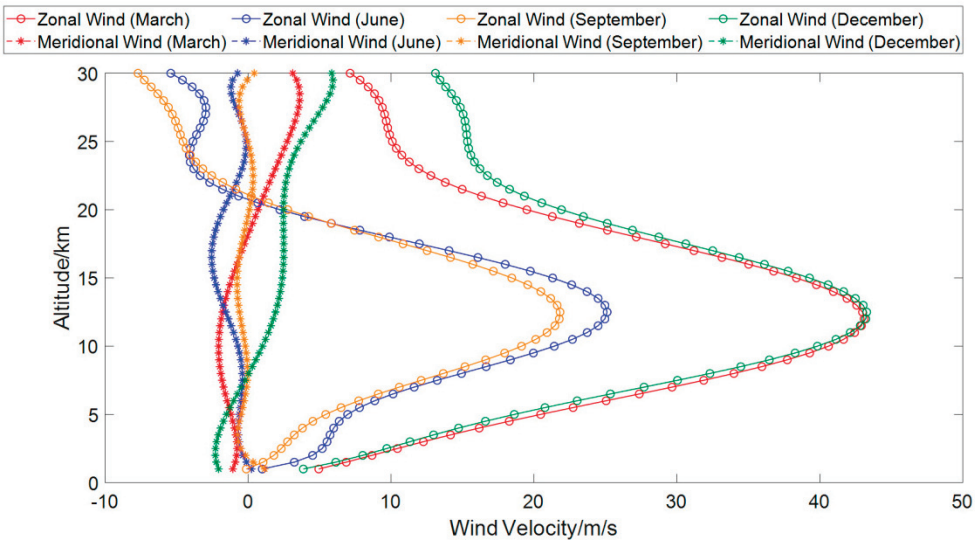


Figure 4. Wind fittings from 1 km to 30 km for a specific region.

From Figure 4, it can be seen that wind velocity changes remarkably with altitude, which provides opportunities for IPBs to adjust their horizontal trajectories utilizing different winds. In particular, a so-called "quasi-zero wind layer" existed at an altitude of about 21 km in June and September, as shown in Figure 4. At altitudes up and down the quasi-zero wind layer, the direction of zonal wind reverses, and the magnitude of meridional wind is small, which is beneficial to IPBs to lengthen their flight time in specific airspace [48–53].

5. Constraints of IPCB Geometry Configuration

IPCB geometry configuration faces many constraints because of its unique features and management strategy, such as airspace constraints, flight altitude constraints, and time interval constraints.

5.1. Airspace Constraint

Airspace is the space in which an IPCB is approved to fly at a certain time. Within the approved airspace, an IPB can fly with its gondola, and its payloads, such as signal generators and transmitters, can run normally. Once an IPB flies out of the approved airspace, its gondola will be cut off from its balloon, and the payloads in the gondola will be switched off, making it impossible for the IPB to emit navigation signals. So, the longitudes and latitudes of each IPB in the IPCB should vary depending on the extent of the approved airspace.

5.2. Flight Altitude Constraint

As discussed above, the flight altitude adjustment of an IPB is achieved by changing its air ballonet volume, which cannot change infinitely. The air ballonet volume of an IPB can only vary in a feasible range, and so does its flight altitude. During IPCB geometry configuration planning, the flight altitudes of all the IPBs should not go beyond the range.

5.3. Time Interval Constraint

Due to the low-density atmosphere at IPCB flight altitudes and the limited capacity of fans, IPCB geometry configuration adjustment requires a long time. So, the time interval between adjacent adjustment actuations should be greater than the maximum trajectory adjustment time required by all the IPBs in the constellation. In this paper, Equations (14) and (15) are used to estimate the time required for an IPB trajectory adjustment.

6. Planning Algorithm of IPCB Geometry Configuration

Based on the analysis above, it can be inferred that if fans and valves can be controlled properly, IPCB geometry configuration can be achieved by utilizing winds at different altitudes effectively. So, the IPCB geometry configuration planning problem can be considered a flight altitude combination problem that can be solved by heuristic algorithms.

WOA is a famous heuristic algorithm that has achieved success in many applications because of its strong robustness, effective searchability, and convenient parameter settings [54–58]. Compared with PSO (particle swarm optimization) or DE (differential evolution algorithm), WOA does not consider subjective parameter settings, such as the inertial coefficient, acceleration coefficient, and other parameters in PSO, scale factor, crossover probability, and other parameters in DE [59,60]. In addition, the performance of WOA, PSO, and DE are compared in reference [54], and WOA displays its excellent capability [54]. Therefore, this paper adopts WOA to realize IPCB geometry configuration planning.

In the planning algorithm, the flight altitudes of all the IPBs in an IPCB can be treated as a whale agent. The general procedure of the algorithm can be described as follows:

Initialize all the whale agents in the current whale population randomly (i.e., initialize the flight altitudes of all the IPBs randomly);

Acquire horizontal winds corresponding to the flight altitudes (i.e., the whale agents just initialized) by Equation (17) or from other data sources;

Calculate horizontal trajectories of all the IPBs in the IPCB by Equation (16);

Adjust the flight altitudes and horizontal trajectories of the IPCB by approaches defined in Section 4;

If the flight altitudes, horizontal trajectories, or adjusted time intervals (calculated by Equation (14) or (15)) do not meet the constraints listed in Section 5, the fitness of the whale agent is defined as 0, meaning that the corresponding IPCB geometry configuration is not feasible in the assumed conditions;

To a whale agent complying with the constraints listed in Section 5, calculate its fitness by Equation (9);

Calculate the fitness of all the whale agents and select the best whale agent in the current whale population;

Update all the whale agents in the current whale population by strategies defined in WOA, such as the "encircling prey" strategy, "bubble-net attacking" strategy, and "search for prey" strategy [54];

Implement updating iteration according to the procedure of WOA, which has been described in detail in reference [54];

When the iteration ends, an IPCB flight altitude can be obtained from the best whale agent. The flight trajectory and constellation GDOP can also be calculated from the best whale agent, forming a complete IPCB geometry configuration.

7. Simulations and Discussions

To verify the effect of the proposed algorithm, simulations are carried out in Matlab 2018b, with the context of IPCB providing independent regional positioning services.

7.1. Simulation Settings

The parameters used in the simulations are listed in Table 1.

Table 1. Parameters used in simulations.

Symbol	Physical Meaning	Value
n_p	number of IPBs in a constellation initially	6
n_t	expected service duration of an IPCB	24 h
c_m	meridional wind coefficients	−0.1338, 1.3189, −1.9669, −2.3772, 4.0187, 1.4032, −1.4290, −0.5504
c_z	zonal wind coefficients	2.1927, −7.6660, −3.0280,28.8161, −2.6879, −41.5979, 2.0248, 21.8084
λ_{min}	minimal longitude of the approved airspace	107° E
λ_{max}	maximal longitude of the approved airspace	109° E
ϕ_{min}	minimal latitude of the approved airspace	39° N
ϕ_{max}	maximal latitude of the approved airspace	41° N
h_{min}	feasible minimal flight altitude of an IPB	20 km
h_{max}	feasible maximal flight altitude of an IPB	24 km

The initial geometry configuration of the IPCB is listed in Table 2, as Figure 5a,b illustrates. Users are distributed uniformly in the service area, as Figure 5c illustrates. The average GDOP of the IPCB with initial geometry configuration is 7.47, whose distribution is illustrated in Figure 5d.

Table 2. Initial geometry configuration of the IPCB.

IPB	Longitude	Latitude	Flight Altitude
IPB 1	107.4° E	39.4° N	21 km
IPB 2	108.6° E	39.4° N	21 km
IPB 3	108.6° E	40.6° N	21 km
IPB 4	107.4° E	40.6° N	21 km
IPB 5	107.8° E	40° N	22 km
IPB 6	108.2° E	40° N	22 km

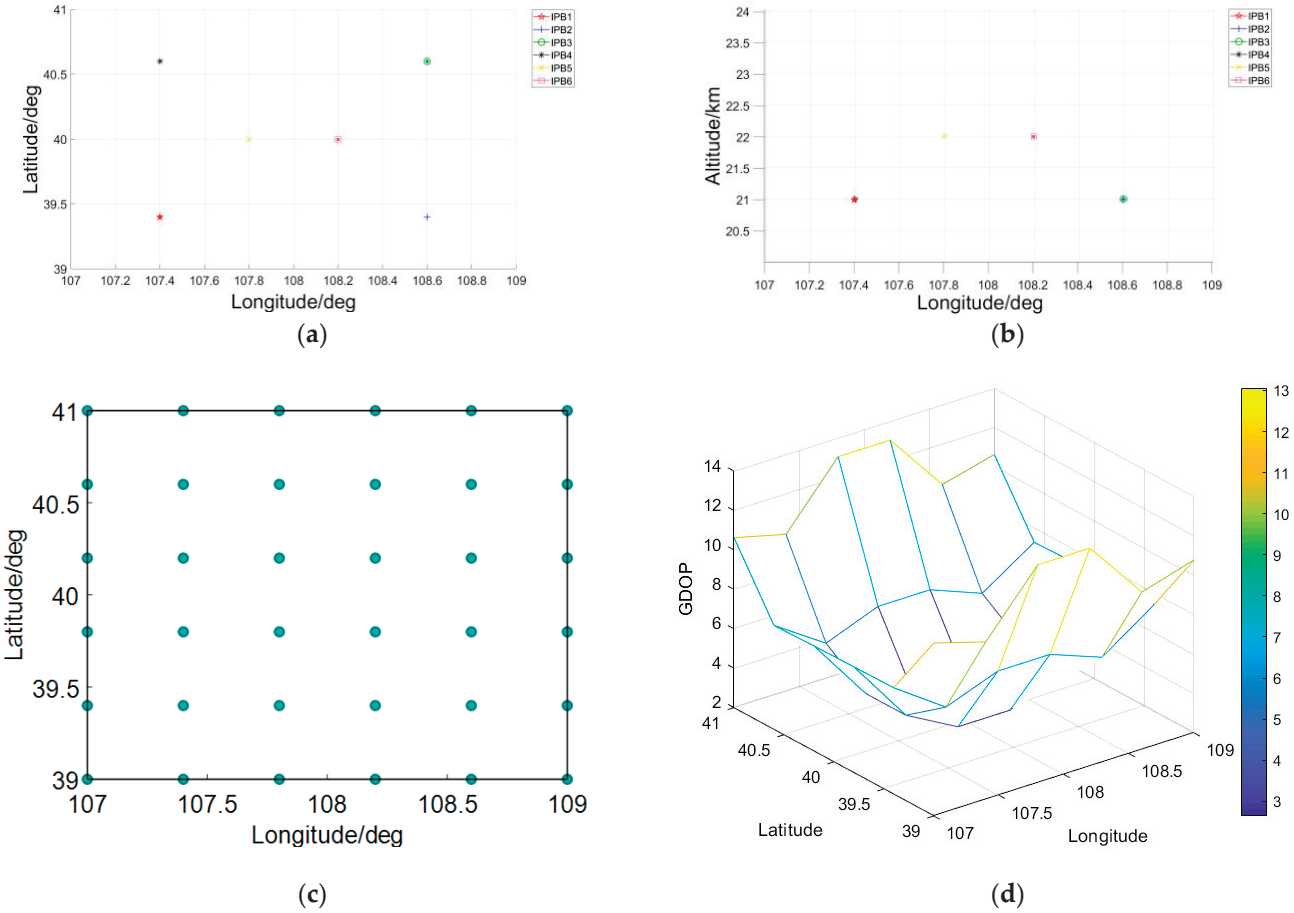


Figure 5. Initial geometry configuration, GDOP distribution of the IPCB, and user distribution: (**a**) initial horizontal layout of the IPCB; (**b**) initial flight altitude of the IPCB; (**c**) user distribution in the service area; (**d**) GDOP distribution of the IPCB with initial geometry configuration.

7.2. Simulation Result

Simulations are carried out with parameters defined in Section 7.1. Figures 6–8 illustrate the planning result. If an IPB flies out of the approved airspace, its subsequent data will not be displayed in figures.

From Figures 6–8, it can be seen that the proposed planning algorithm takes two measures to improve IPCB geometry configuration.

The first measure is to adjust IPB flight altitudes by changing their masses. It makes most IPBs in the constellation fly at an altitude of around 21 km, where wind velocity is small. This measure lengthens IPBs' flight time in the approved airspace, benefiting from keeping the number of available IPBs in the IPCB.

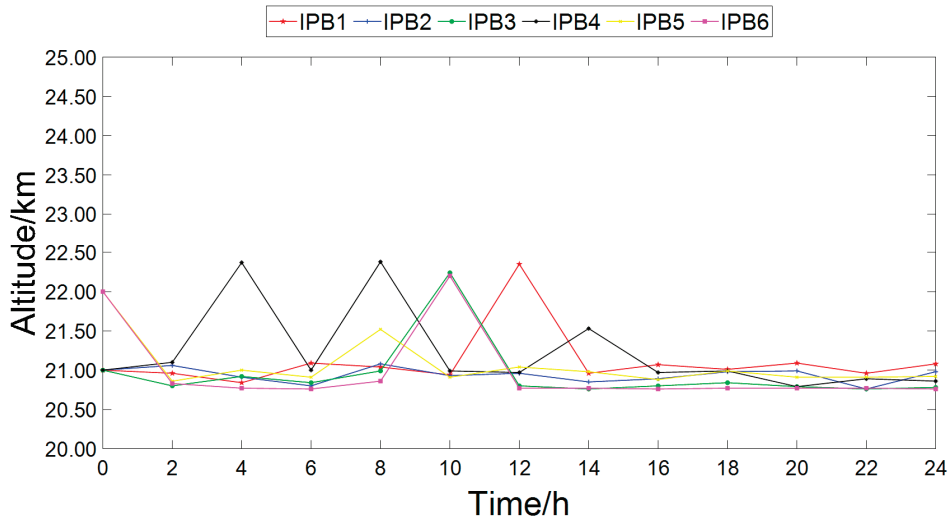


Figure 6. Flight altitude of the IPCB in its service duration.

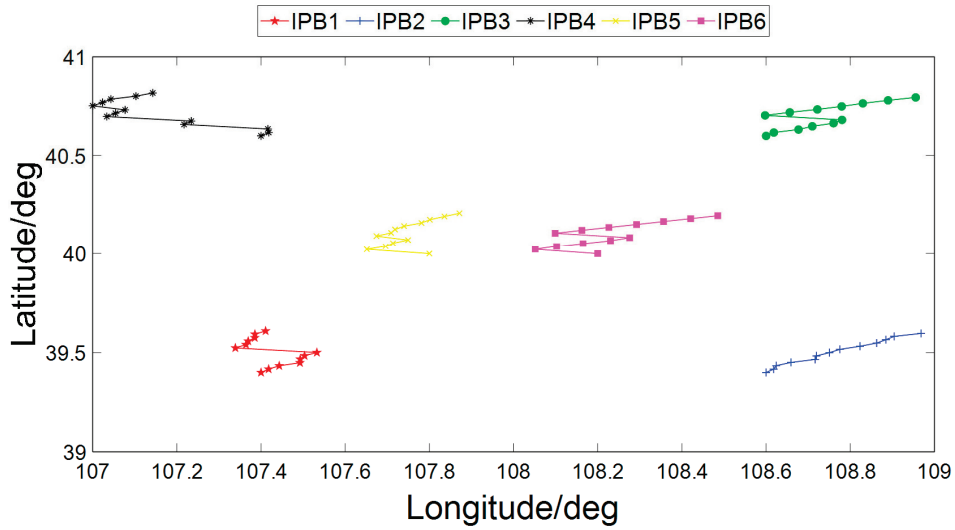


Figure 7. Horizontal trajectory of the IPCB in its service duration.

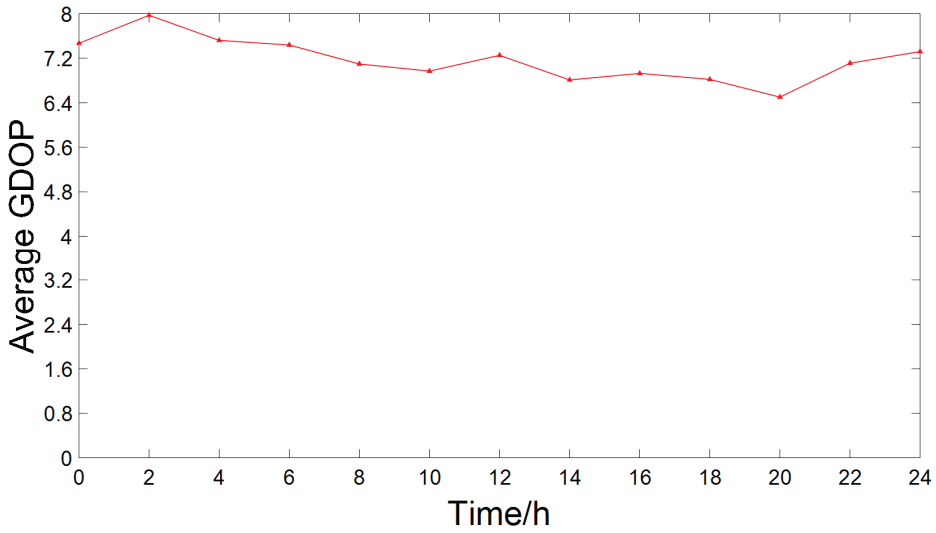
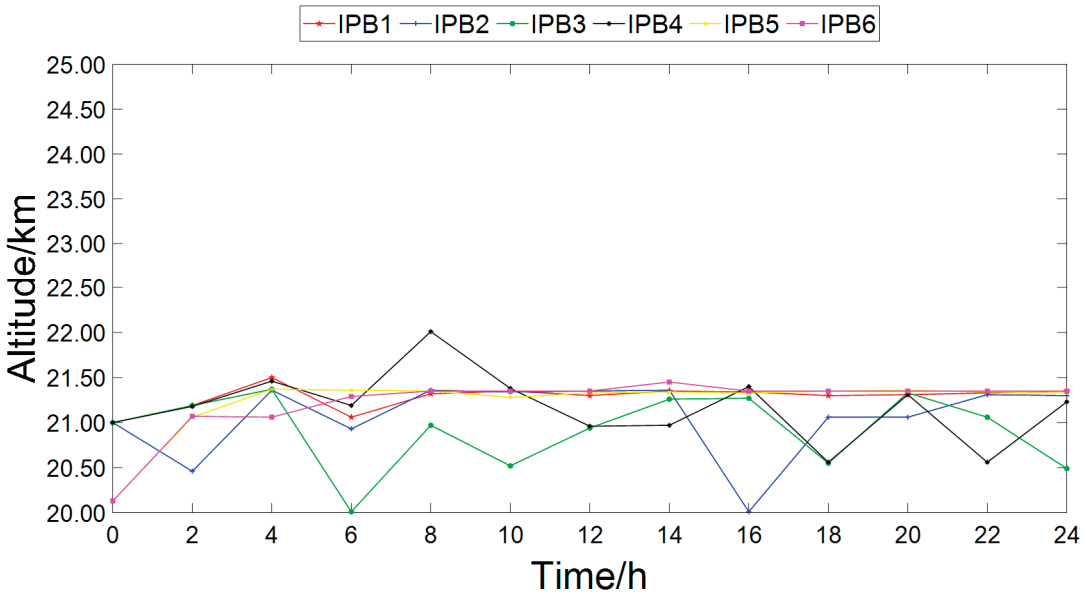


Figure 8. GDOP of the IPCB in its service duration.

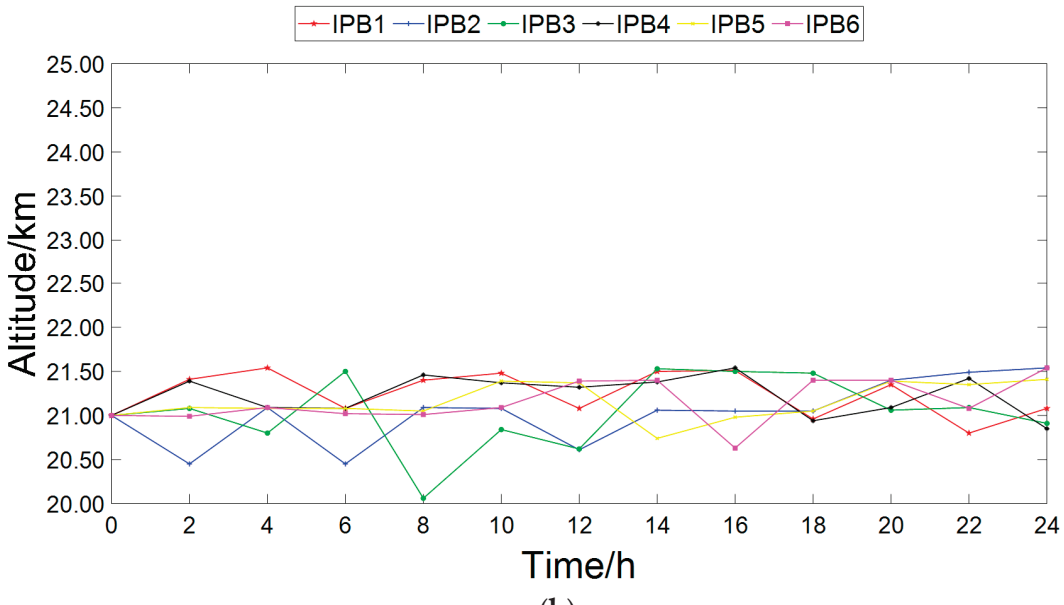
The second measure is to make IPBs fly across the quasi-zero wind layer. This measure utilizes the reverse zonal wind direction to change IPBs' movement direction, which can extend IPBs' flight time in the approved airspace and adjust IPBs' horizontal trajectories, thus improving IPCB geometry configuration.

7.3. Discussion about IPCBs with Different Initial Flight Altitudes

IPBs' initial flight altitudes have a significant impact on IPCB geometry performance. In this section, the initial flight altitudes of IPB5 and IPB6 in Table 2 are modified to 20 km, 21 km, 22 km, 23 km, and 24 km, respectively, while other conditions remain untouched. The planning results of different IPCBs are listed in Figures 9–11.

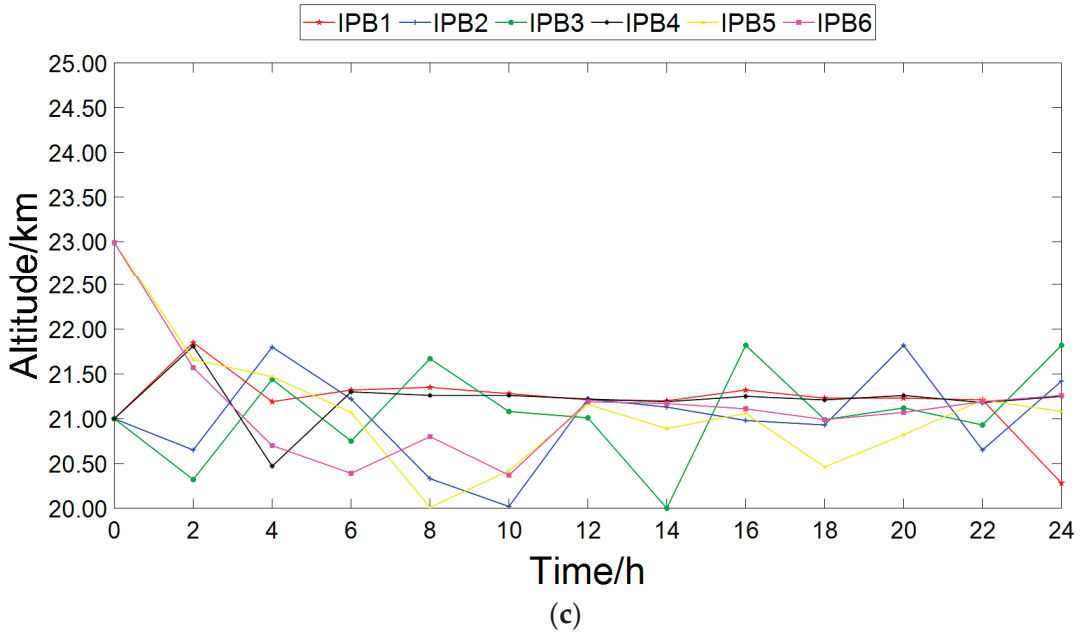


(**a**)

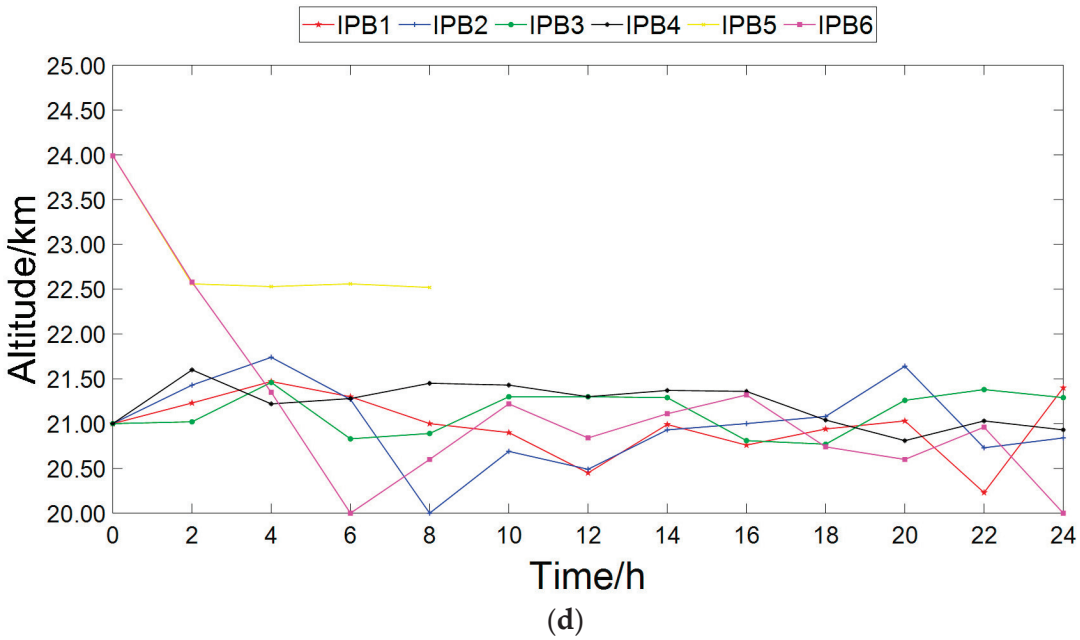


(**b**)

Figure 9. *Cont.*



(**c**)



(**d**)

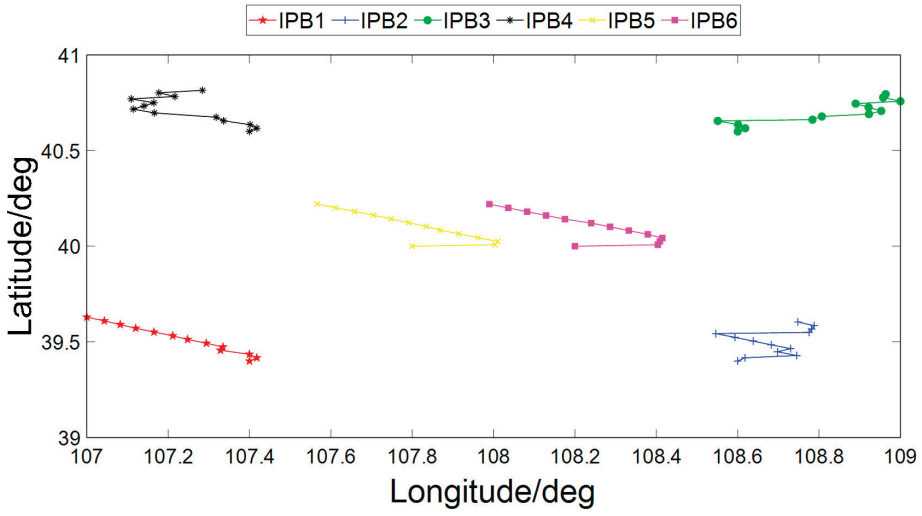
Figure 9. Flight altitude comparison for IPCBs with different initial flight altitudes: (**a**) IPB5 and IPB6 at 20 km initially; (**b**) IPB5 and IPB6 at 21 km initially; (**c**) IPB5 and IPB6 at 23 km initially; (**d**) IPB5 and IPB6 at 24 km initially. (Flight altitude for IPCB with IPB5 and IPB6 at 22 km initially, please refer to Figure 6).

From the comparison of Table 3, it can be seen that there are no significant performance differences in 20 km, 21 km, 22 km, and 23 km, but there is a declining tendency in 24 km. The result of 24 km can probably be attributed to the short service time of IPB5 caused by the big wind velocity. All of these are reflected in Figures 9–11.

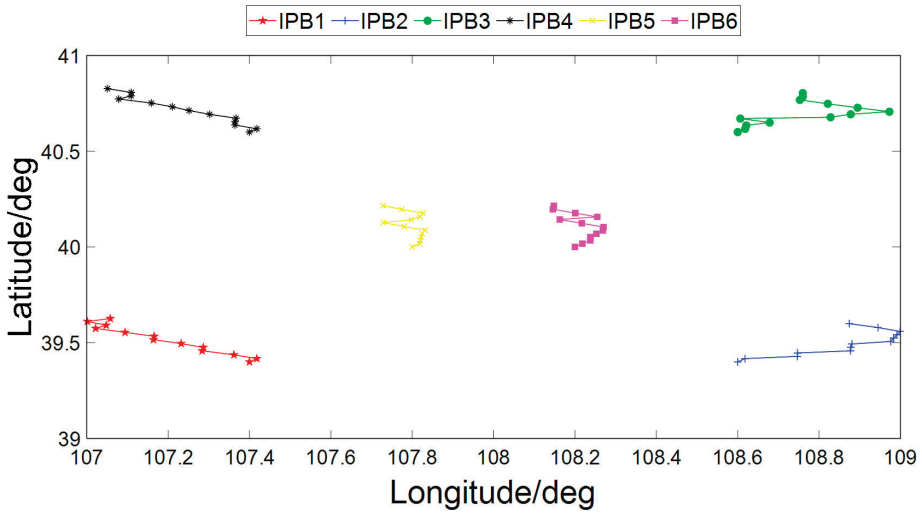
Table 3. Simulation results for IPCBs with different initial flight altitudes.

Initial Flight Altitude of IPB5 and IPB6/km	Average GDOP
20	7.26
21	7.14
22	7.32
23	7.47
24	8.64

In addition, from Figure 11, it can be seen that for the listed initial flight altitudes, the higher the initial flight altitude is, the better the initial GDOP is. However, the IPCB with the highest initial flight altitude has the fastest performance deterioration due to the big wind velocity at the highest altitude. Therefore, if short-term performance is pursued, higher initial flight altitudes may be preferred over lower altitudes. If long-term performance is pursued, initial flight altitudes near the quasi-zero wind layer may be preferred.

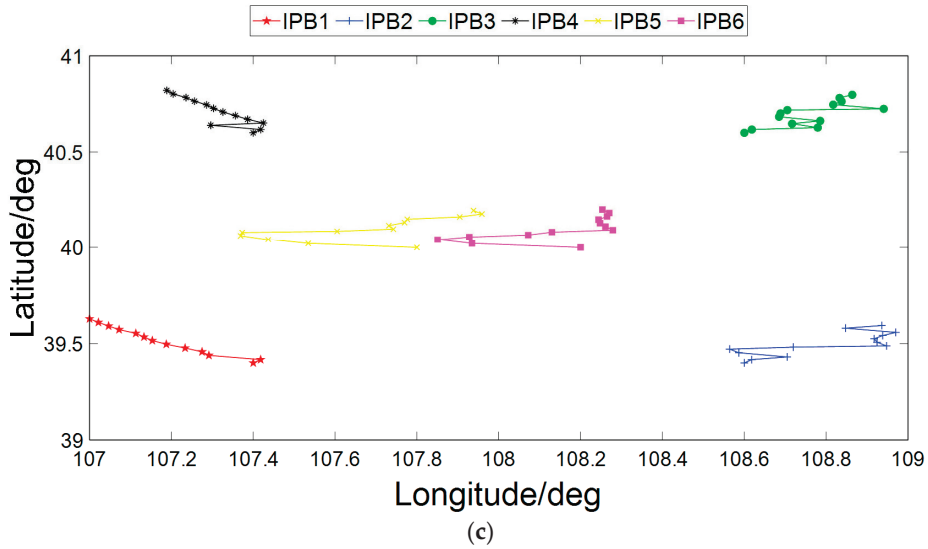


(**a**)

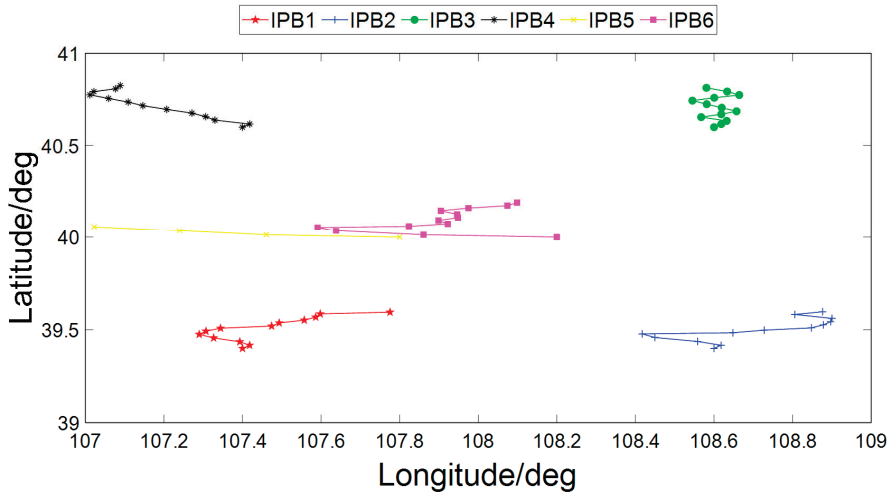


(**b**)

Figure 10. *Cont.*



(**c**)



(**d**)

Figure 10. Horizontal trajectory comparison for IPCBs with different initial flight altitudes: (**a**) IPB5 and IPB6 at 20 km initially; (**b**) IPB5 and IPB6 at 21 km initially; (**c**) IPB5 and IPB6 at 23 km initially; (**d**) IPB5 and IPB6 at 24 km initially. (Horizontal trajectory for IPCB with IPB5 and IPB6 at 22 km initially, please refer to Figure 7).

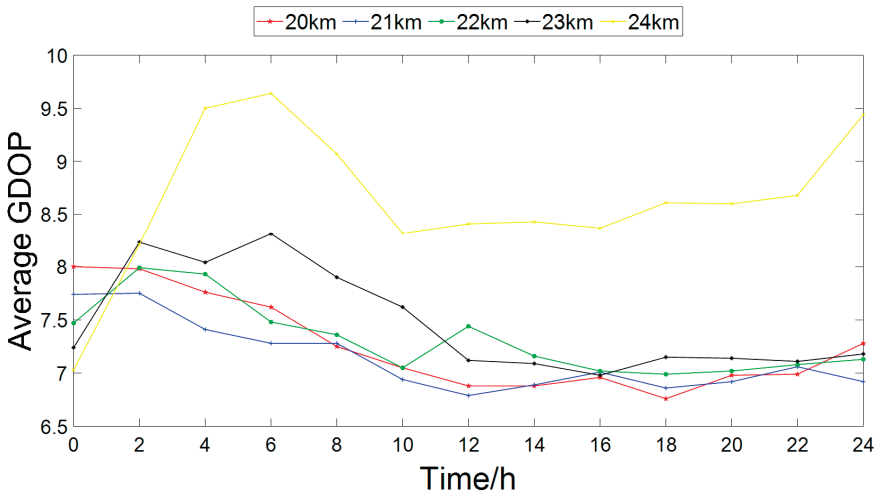
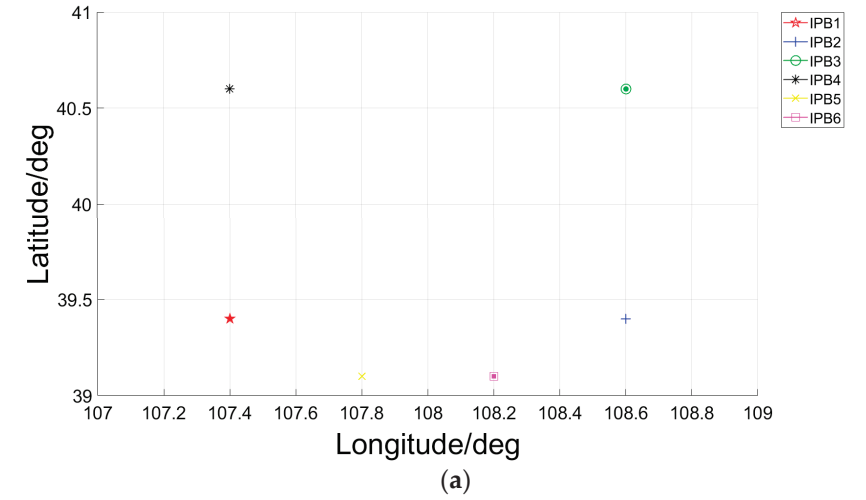


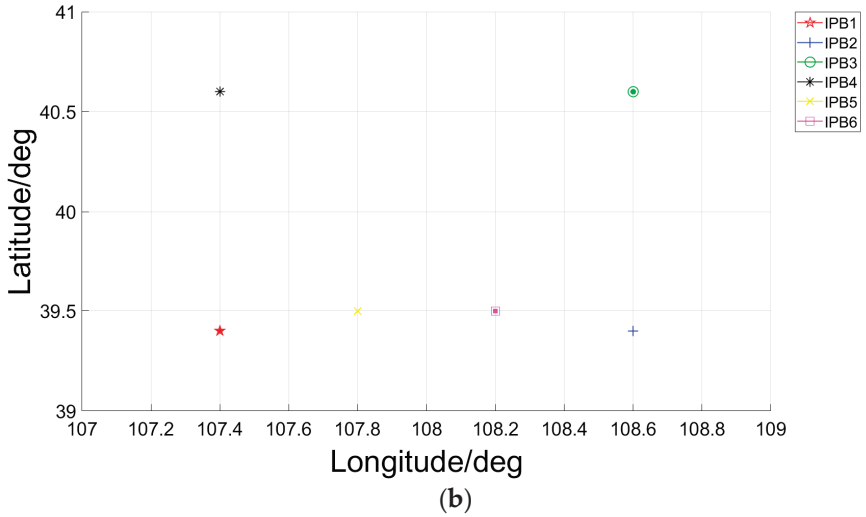
Figure 11. GDOP comparison for IPCBs with different initial flight altitudes.

7.4. Discussion about IPCBs with Different Initial Horizontal Layouts

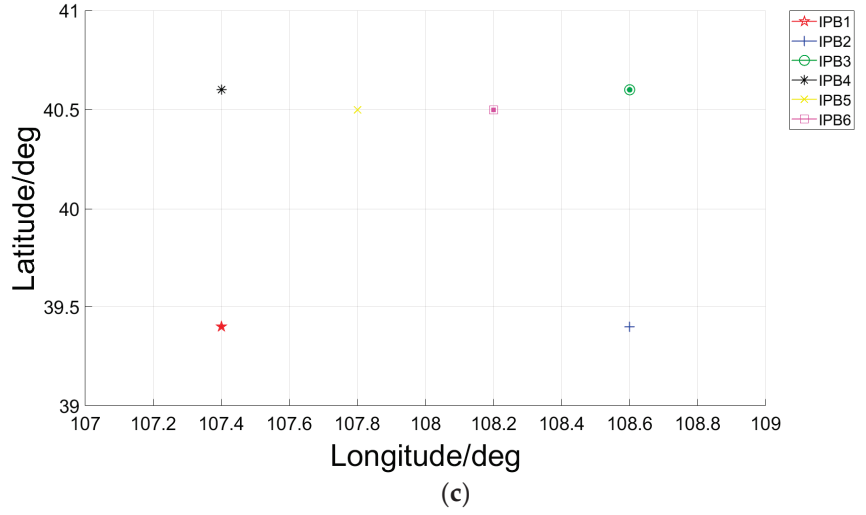
The initial horizontal layout of an IPCB also has a significant impact on its geometry performance. This section adjusts the initial latitude of IPB5 and IPB6 from south to north, as illustrated in Figure 12, while other conditions remain untouched. The planning results are compared in Figures 13–15.



(**a**)

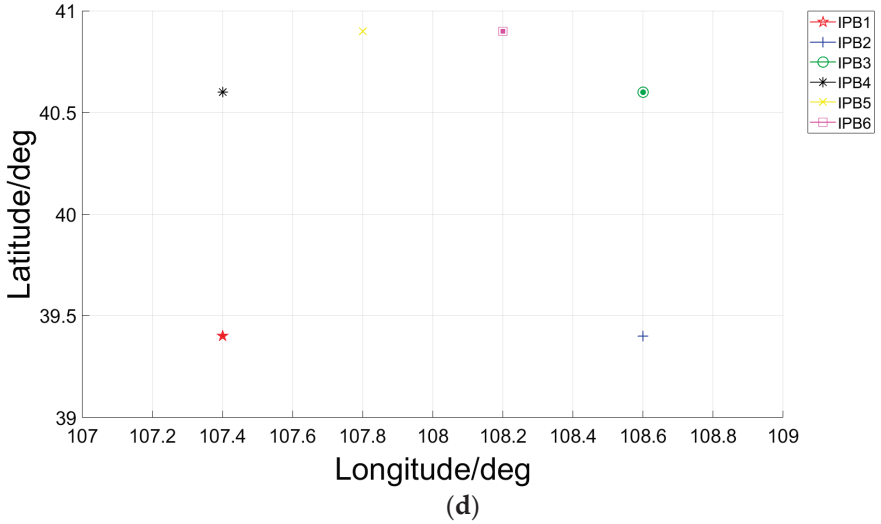


(**b**)



(**c**)

Figure 12. *Cont.*



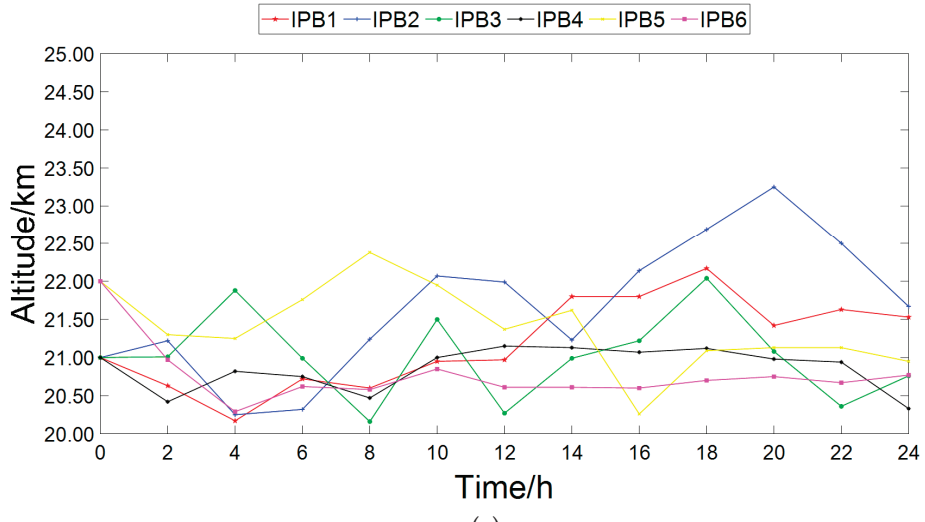
(**d**)

Figure 12. Illustration of different initial horizontal layouts: (**a**) Layout 1 (IPB5 and IPB6 at 39.1° N); (**b**) Layout 2 (IPB5 and IPB6 at 39.5° N); (**c**) Layout 4 (IPB5 and IPB6 at 40.5° N); (**d**) Layout 5 (IPB5 and IPB6 at 40.9° N). (Layout 3 (IPB5 and IPB6 at 40° N), please refered to Figure 5a).

Figures 13–15 and Table 4 show that IPCB can obtain better performance in deploying IPBs with high altitudes to positions near the airspace center (layout 3) than to positions near airspace borders (Layout 1 and Layout 5).

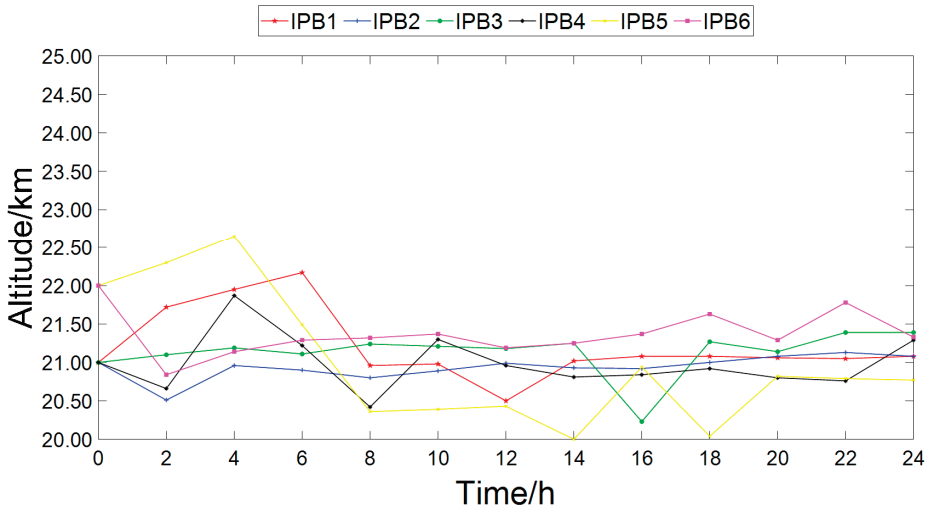
Table 4. Simulation results of IPCBs with different initial horizontal layouts.

Initial Horizontal Layout	Average GDOP
Layout 1	10.88
Layout 2	9.53
Layout 3	7.32
Layout 4	10.83
Layout 5	22.14

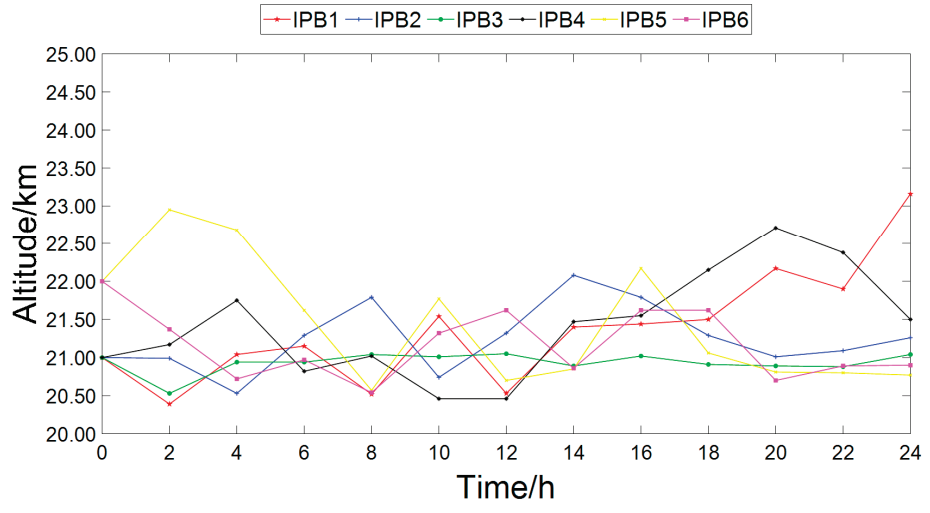


(**a**)

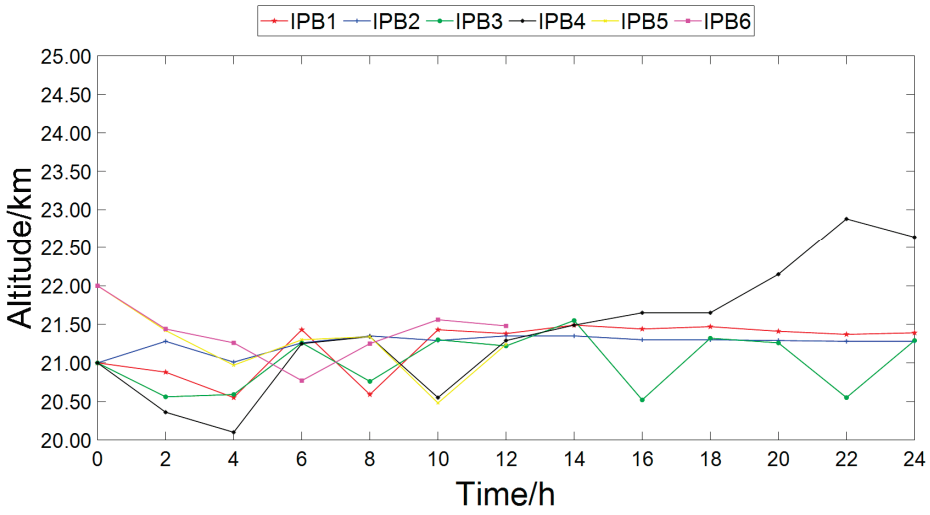
Figure 13. *Cont.*



(**b**)



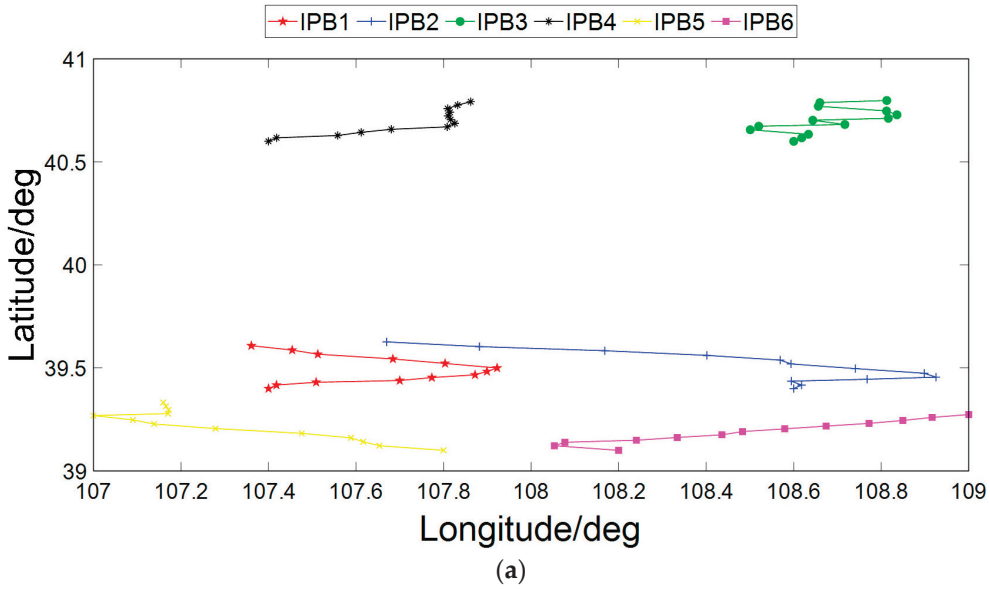
(**c**)



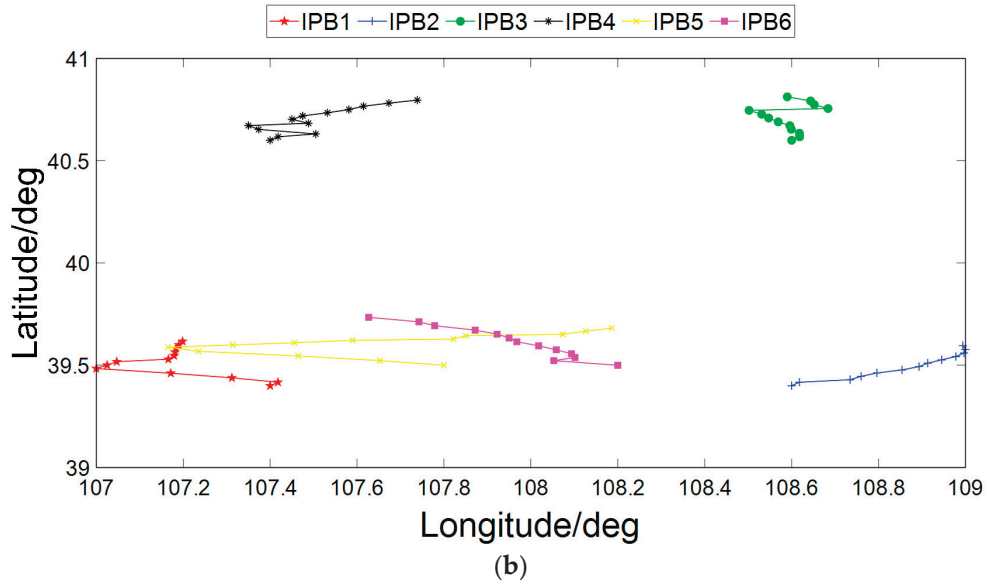
(**d**)

Figure 13. Flight altitude comparison for IPCBs with different initial horizontal layouts: (**a**) Layout 1; (**b**) Layout 2; (**c**) Layout 4; (**d**) Layout 5. (Flight altitude for IPCB with initial Layout 3, please refer to Figure 6).

Furthermore, Figure 15 shows that IPCBs with Layout 1 and Layout 5 have approximate initial GDOPs. However, their performance displays a different tendency as time goes on. A similar case occurs in IPCBs with Layout 2 and Layout 4. Data in Table 5 also shows that Layout 1 performs better than Layout 5, and Layout 2 performs better than Layout 4. These phenomena may be derived from the winds used in the simulations. In the feasible flight altitude range (21~24 km), the meridional winds are all southerly, leading to all the IPBs moving northward. So, IPCBs with low initial latitudes perform better than IPCBs with high initial latitudes. It is especially obvious in Layout 5, in which IPB5 and IPB6 fly out of the approved airspace quickly due to the short distance between their initial positions and the north border of the airspace.



(**a**)



(**b**)

Figure 14. *Cont.*

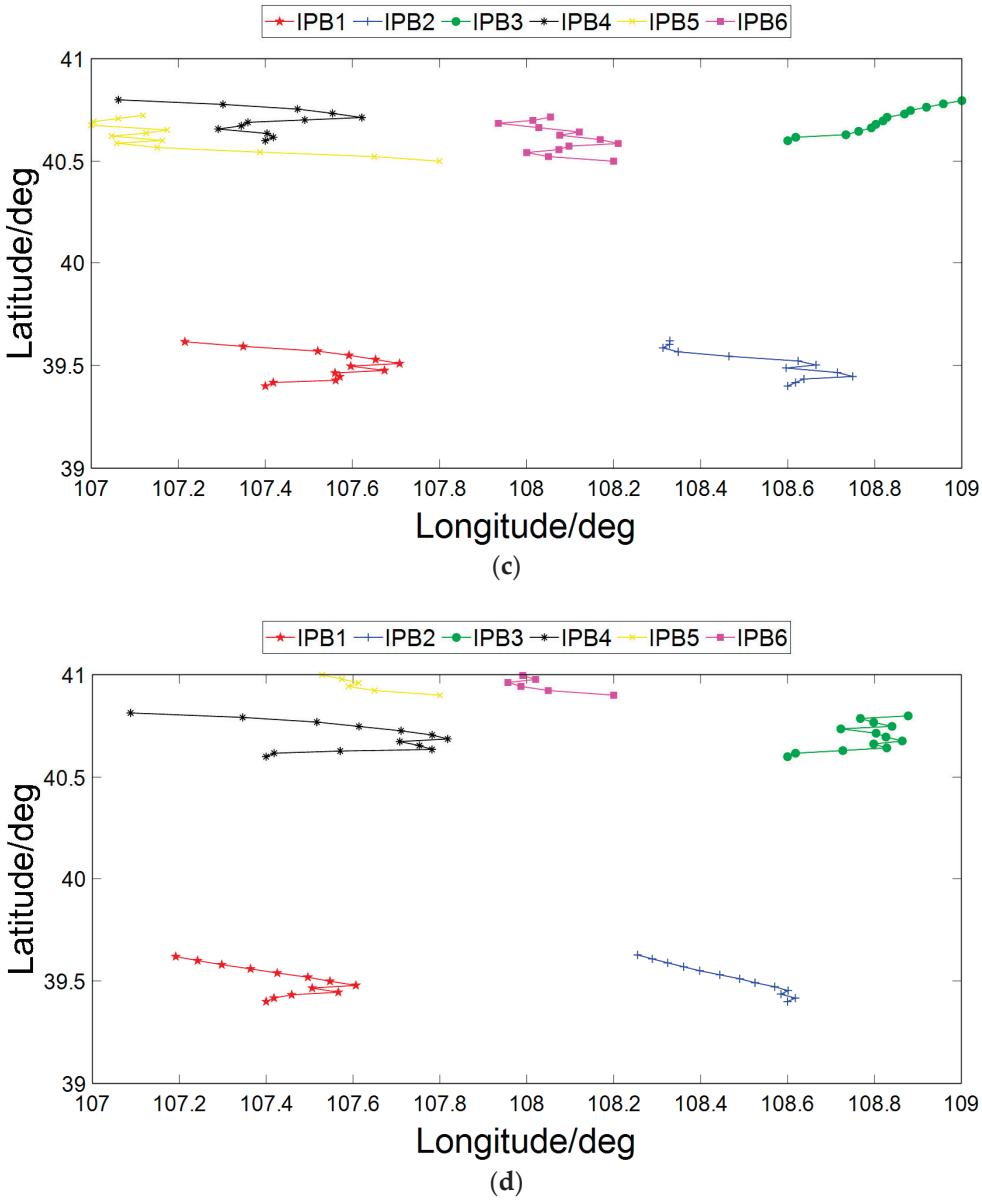


Figure 14. Horizontal trajectory comparison for IPCBs with different initial horizontal layouts: (**a**) Layout 1; (**b**) Layout 2; (**c**) Layout 4; (**d**) Layout 5. (Horizontal trajectory for IPCB with initial Layout 3, please refer to Figure 7).

In winds with quasi-zero wind layers, the direction reversion rule of zonal wind can be employed to improve IPCB geometry configuration. In contrast, no proper rule of meridional wind can be employed. To achieve good performance throughout the whole service duration, the wind is suggested to be treated as a notable factor in the initial horizontal layout design of IPCB.

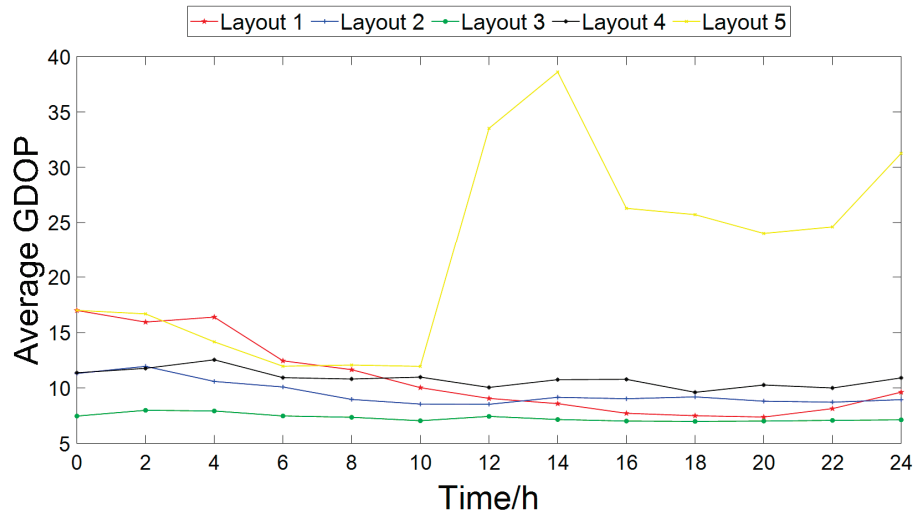


Figure 15. GDOP comparison for IPCBs with different initial horizontal layouts.

Table 5. Coefficients used to fit wind without quasi-zero wind layer.

Symbol	Physical Meaning	Value
c_m	meridional wind coefficients	−0.5066, 4.3171, −14.6477, 27.7923, −36.4969, 32.8921, −13.9287, −0.2023
c_z	zonal wind coefficients	−40.2337, 321.8428, −1018.3275, 1597.1910, −1263.8719, 461.2160, −94.6881, 40.2845

7.5. Discussion about IPCBs in Winds with/without Quasi-Zero Wind Layer

From previous discussions, it can be seen that the quasi-zero wind layer plays an important role in IPCB geometry configuration. However, the quasi-zero wind layer does not always exist. In this section, wind without a quasi-zero wind layer is used to implement the planning. The wind coefficients used in this section are listed in Table 5. The comparison of wind in this section and wind in Section 7.2 is illustrated in Figure 16. The planning result is shown in Figures 17–19.

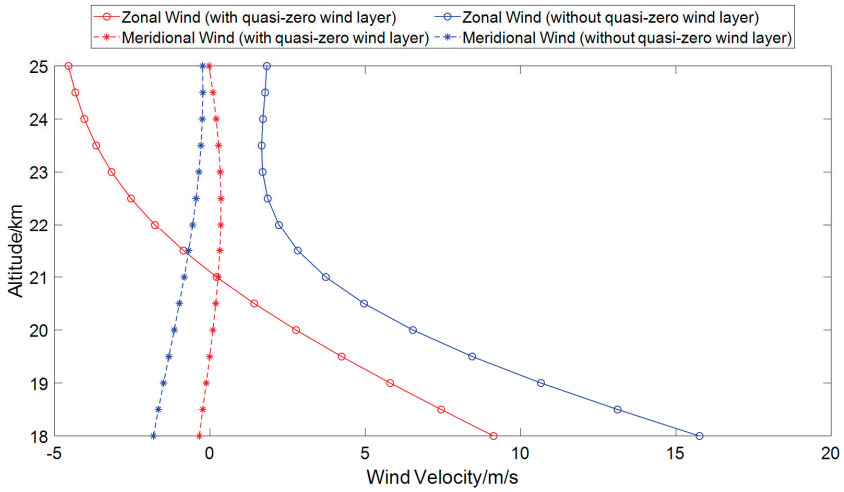


Figure 16. Wind comparison (with/without quasi-zero wind layer).

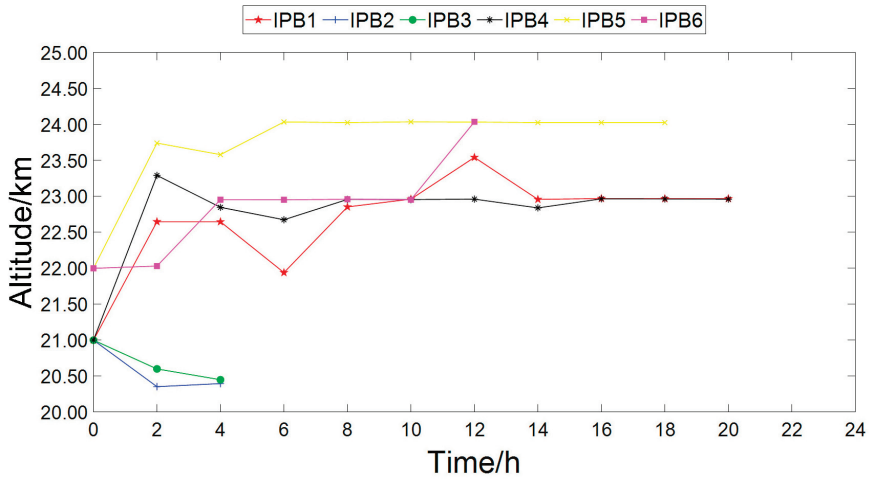


Figure 17. Flight altitude of the IPCB in the wind without a quasi-zero wind layer.

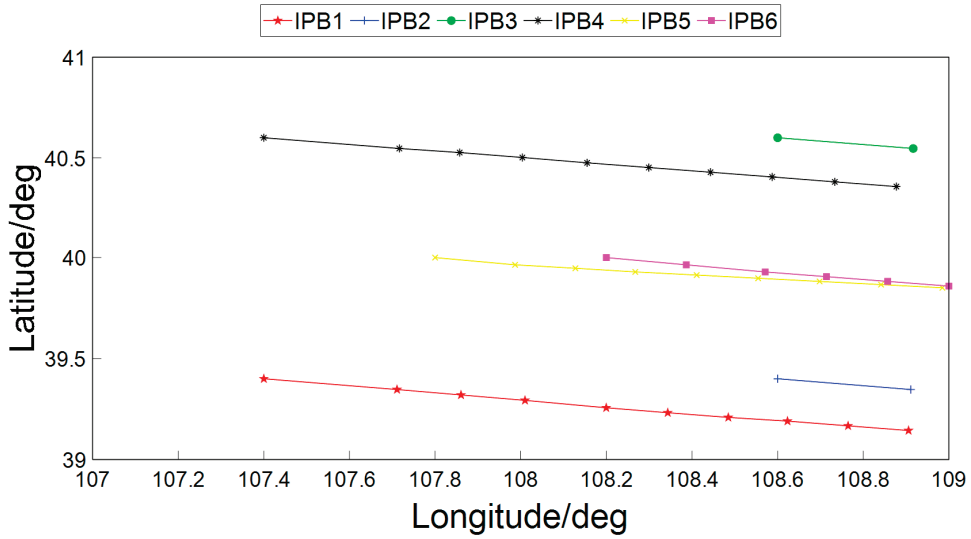


Figure 18. Flight trajectory of the IPCB in the wind without a quasi-zero wind layer.

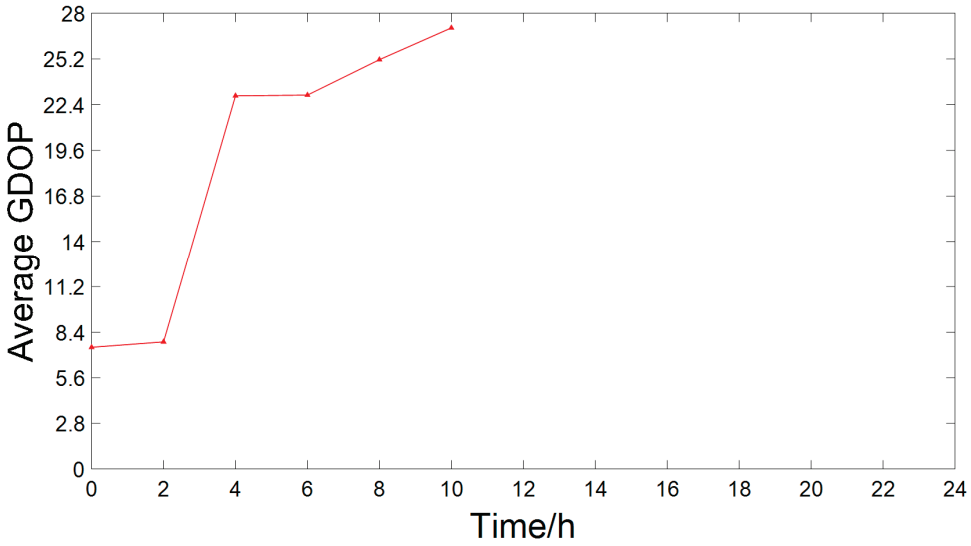


Figure 19. GDOP comparison for IPCBs in winds with/without quasi-zero wind layers.

From Figures 17–19, it can be seen that in winds without a quasi-zero wind layer, zonal trajectory direction reversion does not occur since wind direction reversion does not occur. In addition, the wind velocity is bigger than the wind velocity in Section 7.2, so some IPBs fly out of the approved airspace in a short time, such as IPB2 and IPB3 in Figure 18. This leads to a decrease in the number of available IPBs in the IPCB and deterioration in the IPCB geometry configuration. In order to lengthen IPBs' flight time in the approved airspace, the planning algorithm tends to adjust IPBs' flight altitude to 23~24 km, where wind velocity is small, as IPB1, IPB 4, IPB 5, and IPB6 in Figure 17 illustrate.

By comparing planning results in this section with results in Section 7.2, it can be seen that the results in winds with quasi-zero wind layers are significantly better than those in winds without quasi-zero wind layers. It indicates the significance of the quasi-zero wind layer in improving IPCB geometry performance.

To improve IPCB performance in winds without quasi-zero wind layers, measures such as altering the initial layout, increasing the initial number of IPBs, or supplementing IPBs dynamically can be taken.

Since IPCB achieves its flight mainly by buoyancy and winds, it is sensitive to its running environment. Environment fluctuations or different environment models may bring different results, which can be analyzed in detail in the future.

7.6. Discussion about Uncertain Environment

The simulations above are based on the assumption of a deterministic environment in which wind velocities and atmospheric density are steady. However, uncertain and unknown factors exist in the IPCB running environment.

Conway has investigated horizontal velocity in the midlatitude stratosphere using the observation data of Project Loon and has found the existence of horizontal wind velocity perturbations [61]. Wolf has proposed a model for modeling uncertain winds, which employed a Von Mises distribution to simulate the direction of uncertain wind and a Gaussian distribution to simulate the magnitude of uncertain wind [39]. However, in general, direct observations of winds in the stratosphere are sparse, so a precise model of stratospheric wind is very challenging.

Since this paper focuses on the problem of IPCB geometry configuration, we will not discuss the environment model in detail.

An uncertain environment may degrade the effect of the proposed algorithm because the uncertainty may make IPBs' trajectories deviate from expectation, and as a result, the geometry configuration of IPCB cannot reach the ideal state.

To improve robustness, data filters or environment prediction models can be developed. When perceiving uncertain factors, data filters and prediction models can help to filter out data errors and keep real environment changes. The IPCB can update the environment model and implement the planning algorithm with the new environment model.

8. Conclusions

IPCBs are a novel pseudolite application with many advantages and unique features. Compared with traditional ground-based pseudolites and other air-based pseudolites, IPCB uses high-altitude balloons to achieve higher altitudes and wider coverage. Compared with pseudolites based on powered platforms, IPCBs can save energy costs greatly by utilizing buoyancy and wind. When bringing advantages to applications, these features also bring great challenges to IPCB geometry configuration.

This paper proposes an IPCB geometry configuration planning algorithm that considers the unique features of the IPCB and implements simulations to verify the effectiveness of the proposed algorithm. Furthermore, this paper implements simulations with some typical IPCB geometry configurations and compares their performances.

Simulations show that, in the vertical direction, it can achieve better performance to deploy IPCBs at the altitude of the local quasi-zero wind layer; if the expected service duration is short, IPCBs can be deployed at higher altitudes, and if the expected service

duration is long, IPCBs can be deployed at the altitude of local quasi-zero wind layer. In the horizontal direction, the direction of local wind should be treated as an important factor in designing the initial constellation geometry configuration. A quasi-zero wind layer is helpful in improving IPCB geometry performance.

In the future, attention can be paid to approaches to enhance robust algorithms and enhance tolerance to different environment models. Improvements in algorithm performance are also desired to realize real-time control.

Author Contributions: Conceptualization, Y.Q.; methodology, Y.Q. and S.W.; software, Y.Q.; validation, T.P. and H.F.; writing—original draft preparation, Y.Q.; writing—review and editing, S.W. and T.P.; visualization, Y.Q. and T.P.; supervision, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key Research and Development Program of China (Grant No. 2022YFB3901805).

Data Availability Statement: Data are contained within the article.

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

References

1. Klein, D.; Parkinson, B.W. The use of pseudo-satellites for improving GPS performance. *Navig. J. Inst. Navig.* **1984**, *31*, 303–315. [CrossRef]
2. Morley, T.; Lachapelle, G. GPS augmentation with pseudolites for navigation in constricted waterways. *Navig. J. Inst. Navig.* **1997**, *44*, 359–372. [CrossRef]
3. Wang, J. Pseudolite applications in positioning and navigation: Progress and problems. *J. Glob. Position. Syst.* **2002**, *1*, 48–56. [CrossRef]
4. Dixon, C.S.; Morrison, R.G. A pseudolite-based maritime navigation system: Concept through to demonstration. *J. Glob. Position. Syst.* **2008**, *7*, 9–17. [CrossRef]
5. Ma, C.; Yang, J.; Chen, J. Satellite-Ground Joint Positioning System Based on Pseudolite. In Proceedings of the 2018 IEEE CSAA Guidance, Navigation and Control Conference, CGNCC, Xiamen, China, 10–12 August 2018. [CrossRef]
6. Sheng, C.; Gan, X.; Yu, B.; Zhang, J. Precise point positioning algorithm for pseudolite combined with GNSS in a constrained observation environment. *Sensors* **2020**, *20*, 1120. [CrossRef]
7. Liu, T.; Liu, J.; Wang, J.; Zhang, H.; Zhang, B.; Ma, Y.; Sun, M.; Lv, Z.; Xu, G. Pseudolites to support location services in smart cities: Review and prospects. *Smart Cities* **2023**, *6*, 2081–2105. [CrossRef]
8. Kayhan, Ö.; Yücel, Ö.; Hastaoğlu, M.A. Simulation and control of serviceable stratospheric balloons traversing a region via transport phenomena and PID. *Aerosp. Sci. Technol.* **2016**, *53*, 232–240. [CrossRef]
9. Jiang, Y.; Lv, M.; Zhu, W.; Du, H.; Zhang, L.; Li, J. A method of 3-D region controlling for scientific balloon long-endurance flight in the real wind. *Aerosp. Sci. Technol.* **2020**, *97*, 1–11. [CrossRef]
10. Zhu, W.; Xu, Y.; Du, H.; Li, J. Thermal performance of high-altitude solar powered scientific balloon. *Renew. Energy* **2019**, *135*, 1078–1096. [CrossRef]
11. Jiang, Y.; Lv, M.; Qu, Z.; Zhang, L. Performance evaluation for scientific balloon station-keeping strategies considering energy management strategy. *Renew. Energy* **2020**, *156*, 290–302. [CrossRef]
12. Li, C.; Luo, R.; Chen, T. New idea for stratospheric communications—Google Loon. *Commun. Technol.* **2015**, *48*, 125–129. [CrossRef]
13. Deng, X.; Yang, X.; Zhu, B.; Ma, Z.; Hou, Z. Simulation research and key technologies analysis of intelligent stratospheric aerostat Loon. *Acta Aeronaut. Et Astronaut. Sin.* **2023**, *44*, 127412. [CrossRef]
14. Rabinowitz, M.; Parkinson, B.W.; Cohen, C.E.; O'Connor, M.L.; Lawrence, D.G. A system using LEO telecommunication satellites for rapid acquisition of integer cycle ambiguities. In Proceedings of the Position Location and Navigation Symposium, Palm Springs, CA, USA, 20–23 April 1996; pp. 137–145. [CrossRef]
15. Li, X.; Li, X.; Ma, F.; Yuan, Y.; Zhang, K.; Zhou, F.; Zhang, X. Improved PPP ambiguity resolution with the assistance of multiple LEO constellations and signals. *Remote Sens.* **2019**, *11*, 408. [CrossRef]
16. Ge, H.; Li, B.; Jia, S.; Nie, Y.; Wu, T.; Yang, Z.; Shang, J.; Zheng, Y.; Ge, M. LEO enhanced global navigation satellite system (LeGNSS): Progress, opportunities, and challenges. *Geo-Spat. Inf. Sci.* **2022**, *25*, 1–13. [CrossRef]
17. Yuan, H.; Chen, X.; Luo, R.; Wan, H.; Zhang, Y.; Li, R.; Yang, G. Review of the development trend of LEO-based navigation system. *Navig. Position. Timing* **2022**, *9*, 1–11. [CrossRef]
18. Zhang, Y.; Fan, L.; Liu, J.; Li, Z. Feasibility analysis of commercial broadband LEO constellation incorporated into the national comprehensive PNT system. *Navig. Position. Timing* **2022**, *10*, 26–36. [CrossRef]

19. Dai, L.; Wang, J.; Tsujii, T.; Rizos, C. Pseudolite Applications in Positioning and Navigation: Modelling and Geometric Analysis. In Proceedings of the International Symposium on Kinematic Systems in Geodesy, Geomatics & Navigation (KIS2001), Banff, AB, Canada, 5–8 June 2001.
20. Chandu, B.; Pant, R.S.; Moudgalya, K. Modeling and Simulation of a Precision Navigation System Using Pseudolites Mounted on Airships. In Proceedings of the 7th AIAA Aviation Technology, Integration and Operations Conference, Belfast, Northern Ireland, 18–20 September 2007.
21. Oktay, H. Airborne pseudolites in a global positioning system degraded environment. In Proceedings of the 5th International Conference on Recent Advances in Space Technologies–RAST2011, Istanbul, Turkey, 9–11 June 2011. [CrossRef]
22. Tsujii, T.; Harigae, M.; Barnes, J.; Wang, J.; Rizos, C. Experiments of inverted pseudolite positioning for airship-based GPS augmentation system. In Proceedings of the 15th International Technical Meeting of the Satellite Division of the U.S. Institute of Navigation, Portland, OR, USA, 24–27 September 2002; pp. 1689–1695.
23. Sultana, Q.; Sunehra, D.; Ratnam, D.V.; Rao, P.S.; Sarma, A.D. Significance of instrumental biases and dilution of precision in the context of GAGAN. *Indian J. Radio Space Phys.* **2007**, *36*, 405–410.
24. Quddusa, S.; Dhiraj, S.; Vemuri, S.S.; Achanta, D.S. Effects of pseudolite positioning on DOP in LAAS. *Positioning* **2010**, *1*, 18–26. [CrossRef]
25. Wang, J.; Li, H.; Lu, J.; Li, K.; Li, H.; Yang, L.; Li, Y. A PSO-Based Layout Method for GNSS Pseudolite System. In Proceedings of the ICIT 2017, Singapore, 27–29 December 2017. [CrossRef]
26. Jiang, M.; Li, R.; Liu, W. Research on geometric configuration of pseudolite positioning system. *Comput. Eng. Appl.* **2017**, *53*, 271–276.
27. Tang, W.; Chen, J.; Yu, C.; Ding, J.; Wang, R. A new ground-based pseudolite system deployment algorithm based on MOPSO. *Sensors* **2021**, *21*, 5364. [CrossRef]
28. Du, H.; Lv, M.; Li, J.; Zhu, W.; Zhang, L.; Wu, Y. Station-keeping performance analysis for high altitude balloon with altitude control system. *Aerosp. Sci. Technol.* **2019**, *92*, 644–652. [CrossRef]
29. Du, H.; Li, J.; Zhu, W.; Qu, Z.; Zhang, L.; Lv, M. Flight performance simulation and station-keeping endurance analysis for stratospheric super-pressure balloon in real wind field. *Aerosp. Sci. Technol.* **2019**, *86*, 1–10. [CrossRef]
30. Du, H.; Lv, M.; Zhang, L.; Zhu, W.; Wu, Y.; Li, J. Energy management strategy design and station-keeping strategy optimization for high altitude balloon with altitude control system. *Aerosp. Sci. Technol.* **2019**, *93*, 1–9. [CrossRef]
31. Song, J.; Hou, C.; Xue, G.; Ma, M. Study of constellation design of pseudolites based on improved adaptive genetic algorithm. *J. Commun.* **2016**, *11*, 879–885. [CrossRef]
32. Tian, R.; Cui, Z.; Zhang, S.; Wang, D. Overview of navigation augmentation technology based on LEO. *Navig. Position. Timing* **2021**, *8*, 66–81. [CrossRef]
33. Kaplan, E.D.; Hegarty, C.J. *Understanding GPS: Principles and Applications, 2nd ed*; Artech House Inc.: Norword, MA, USA, 2006; p. 02062.
34. Teng, Y.; Wang, J.; Huang, Q. Mathematical minimum of geometric dilution of precision (GDOP) for dual-GNSS constellations. *Adv. Space Res.* **2016**, *57*, 183–188. [CrossRef]
35. Nalineekumari, A.; Sasibhushana, R.G.; Ashok, K.N. GDOP analysis with optimal satellite using GA for southern region of Indian subcontinent. *Procedia Comput. Sci.* **2018**, *143*, 303–308. [CrossRef]
36. Teng, Y.; Wang, J. A closed-form formula to calculate geometric dilution of precision (GDOP) for multi-GNSS constellations. *GPS Solut.* **2016**, *20*, 331–339. [CrossRef]
37. Li, D.; Deng, P.; Liu, B.; Qu, Y.; Zeng, L.; Liu, T. Research on the Dynamic Configuration of Air-Based Pseudolite Network. In *China Satellite Navigation Conference (CSNC) 2015 Proceedings*; Springer: Berlin/Heidelberg, Germany, 2015; Volume II, pp. 357–367.
38. Blackmore, B.; Kuwata, Y.; Wolf, M.T.; Assad, C.; Fathpour, N.; Newman, C.; Elfes, A. Global Reachability and Path Planning for Planetary Exploration with Montgolfirer Balloons. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage Convention District, Anchorage, AK, USA, 3–8 May 2010.
39. Wolf, M.T.; Blackmore, L.; Kuwata, Y.; Fathpour, N.; Newman, C. Probabilistic Motion Planning of Balloons in Strong, Uncertain Wind Fields. In Proceedings of the IEEE International Conference on Robotics & Automation, Anchorage, AK, USA, 3–7 May 2010. [CrossRef]
40. Zhai, J.; Yang, X.; Deng, X.; Long, Y.; Zhang, J.; Bai, F. Global path planning of stratospheric aerostat in uncertain wind field. *J. Beijing Univ. Aeronaut. Astronaut.* **2023**, *49*, 1116–1126. [CrossRef]
41. Bellemare, M.G.; Candido, S.; Castro, P.S.; Gong, J.; Machado, M.C.; Moitra, S.; Ponda, S.S.; Wang, Z. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature* **2020**, *588*, 77–82. [CrossRef]
42. Furfaro, R.; Lunine, J.I.; Elfes, A.; Reh, K. Wind-based navigation of a hot-air balloon on Titan: A feasible study. In Proceedings of the SPIE Defense and Security Symposium, Orlando, FL, USA, 18–20 March 2008. [CrossRef]
43. Hedin, A.E.; Biondi, M.A.; Burnside, R.G.; Hernandez, G.; Johnson, R.M.; Killeen, T.L.; Mazaudier, C.; Meriwether, J.W.; Salah, J.E.; Sica, R.J.; et al. Revised global model of thermosphere winds using satellite and ground-based observations. *J. Geophys. Res.* **1991**, *96*, 7657–7688. [CrossRef]
44. Mueller, J.B.; Zhao, Y.J.; Garrard, W.L. Optimal ascent trajectories for stratospheric airships using wind energy. *J. Guid. Control. Dyn.* **2009**, *32*, 1232–1245. [CrossRef]

45. Lee, S.; Bang, H. Three-dimensional ascent trajectory optimization for stratospheric airship platforms in the jet stream. *J. Guid. Control. Dyn.* **2007**, *30*, 1341–1352. [CrossRef]
46. Sun, S.; Li, Z.; Tian, K. Modeling and trajectory planning of return process for a class of airship with wind field. *Aerosp. Control. Appl.* **2014**, *40*, 37–41. [CrossRef]
47. Zhang, X.; Chen, L.; Quan, T. Study of ascent trajectory for stratospheric airship in the jet stream. *Electron. Des. Eng.* **2014**, *22*, 11–13.
48. Belmont, A.D.; Dartt, D.G.; Nastrom, G.D. Variations of stratospheric zonal winds, 20–65 km, 1961–1971. *J. Appl. Meteorol.* **1975**, *14*, 585–594. [CrossRef]
49. Tao, M.; He, J.; Liu, Y. Analysis of the characteristics of the stratospheric quasi-zero wind layer and the effects of the quasi-biennial oscillation on it. *Clim. Environ. Res.* **2012**, *17*, 92–102. [CrossRef]
50. Chang, X.; Bai, Y.; Fu, W.; Yan, J. Research on fixed-point aerostat based on its special stratosphere wind field. *J. Northwestern Polytech. Univ.* **2014**, *32*, 12–17.
51. Deng, X.; Li, K.; Yu, C.; Yang, X.; Hou, Z. Station-keeping performance of novel near-space aerostat in quasi-zero wind layer. *J. Natl. Univ. Def. Technol.* **2019**, *41*, 5–12. [CrossRef]
52. Chen, B.; Liu, Y.; Liu, L.; Shen, X.; Zhang, Y. Characteristics of spatial–temporal distribution of the stratospheric quasi-zero wind layer in low-latitude regions. *Clim. Environ. Res.* **2018**, *23*, 657–669. [CrossRef]
53. Roney, J.A. Statistical wind analysis for near-space applications. *J. Atmos. Sol. Terr. Phys.* **2007**, *69*, 1485–1501. [CrossRef]
54. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
55. Mafarja, M.M.; Mirjalili, S. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [CrossRef]
56. Xiong, G.; Zhang, J.; Shi, D.; He, Y. Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm. *Energy Convers. Manag.* **2018**, *174*, 388–405. [CrossRef]
57. Ling, Y.; Zhou, Y.; Luo, Q. Lévy flight trajectory-based whale optimization algorithm for global optimization. *IEEE Access* **2017**, *5*, 6168–6186. [CrossRef]
58. Yan, Z.; Wang, S.; Liu, B.; Li, X. Application of Whale Optimization Algorithm in Optimal Allocation of Water Resources. In Proceedings of the 2018 3rd International Conference on Advances in Energy and Environment Research (ICAEER 2018), Guilin, China, 10–12 August 2018. [CrossRef]
59. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN95–International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995. [CrossRef]
60. Neri, R.; Tirronen, V. Scale factor local search in differential evolution. *Memetic Comp.* **2009**, *1*, 153–171. [CrossRef]
61. Conway, J.P.; Bodeker, G.E.; Waugh, D.W.; Murphy, D.J.; Cameron, C.; Lewis, J. Using project Loon superpressure balloon observations to investigate the inertial peak in the intrinsic wind spectrum in the midlattitude stratosphere. *J. Geophys. Res. Atmos.* **2019**, *124*, 8594–8604. [CrossRef]

electronics

MDPI

Article

Research on Aspect-Level Sentiment Analysis Based on Adversarial Training and Dependency Parsing

Erfeng Xu [1,2], Junwu Zhu [1,*], Luchen Zhang [3,*], Yi Wang [1,2] and Wei Lin [1,2]

[1] School of Information Engineering, Yangzhou University, Yangzhou 225127, China; mz120220937@stu.yzu.edu.cn (E.X.); mz220220329@stu.yzu.edu.cn (Y.W.); mx120220559@stu.yzu.edu.cn (W.L.)
[2] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[3] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100190, China
* Correspondence: jwzhu@yzu.edu.cn (J.Z.); zlc@cert.org.cn (L.Z.)

Abstract: Aspect-level sentiment analysis is used to predict the sentiment polarity of a specific aspect in a sentence. However, most current research cannot fully utilize semantic information, and the models lack robustness. Therefore, this article proposes a model for aspect-level sentiment analysis based on a combination of adversarial training and dependency syntax analysis. First, BERT is used to transform word vectors and construct adjacency matrices with dependency syntactic relationships to better extract semantic dependency relationships and features between sentence components. A multi-head attention mechanism is used to fuse the features of the two parts, simultaneously perform adversarial training on the BERT embedding layer to enhance model robustness, and, finally, to predict emotional polarity. The model was tested on the SemEval 2014 Task 4 dataset. The experimental results showed that, compared with the baseline model, the model achieved significant performance improvement after incorporating adversarial training and dependency syntax relationships.

Keywords: multi head attention mechanism; dependency syntactic relationships; adjacency matrix; adversarial training

1. Introduction

The advent of the Internet and the proliferation of social media platforms have led to an exponential increase in the creation and dissemination of textual content on a daily basis. These data contain rich emotional information, which are crucial for understanding users' attitudes and emotional changes towards products, services, or events. Sentiment analysis, a core component of natural language processing, seeks to automatically discern and extract emotional inclinations from textual data. Its significance has grown notably, finding utility in various sectors including information retrieval, social media analysis, and public opinion monitoring [1,2].

Aspect-level sentiment analysis is a subtask of text sentiment classification. In contrast to general sentiment analysis tasks, which focus on predicting the overall sentiment of a text, aspect-based sentiment analysis tasks necessitate predicting emotional polarity towards specific aspects mentioned within a sentence [3]. Aspect-level sentiment analysis presents a unique challenge wherein different aspect words within the same sentence may exhibit varying emotional polarities. For instance, in the sentence "The food was delicious but the service was bad", "food" and "service" represent distinct aspects. While the evaluation of the food is positive ("delicious"), the evaluation of the service is negative ("bad"). The presence of multi-sentiment scenarios amplifies the complexities inherent in aspect-level sentiment analysis. Models tasked with this challenge must possess the capability to

effectively distinguish between different aspects within a sentence and accurately predict the emotional polarity associated with each aspect.

Conventional methodologies for aspect-level sentiment analysis often employed statistical machine learning methods such as naive Bayes or SVM [4], which typically rely on manually designed features for modeling. While these methods have achieved some success to a certain extent, they often rely heavily on the quality and quantity of feature engineering. and struggle with handling complex semantic information and syntactic structures.

The advancement of deep learning, particularly with the emergence of pre-trained language models, has propelled significant strides in aspect-level sentiment analysis. Models such as BERT, RoBERTa, and XLNet [5–7], trained on extensive datasets using self-supervised learning techniques, offer enhanced modeling capabilities for aspect-level sentiment analysis. Mao et al. conducted an empirical study analyzing biases in pre-trained language models (PLMs) for calculating sentiment analysis and emotion detection tasks [8]. It found that RoBERTa outperforms other PLMs in these tasks and proposed methods to mitigate biases.

However, most current aspect-level sentiment analysis methods based on pretrained language models still have limitations. First, these methods often focus on the overall emotional polarity of a sentence while ignoring the relationships between words. Second, the robustness and generalization ability of these models are relatively limited, and they may lead to incorrect classification when exposed to external perturbations.

To address these shortcomings, the presented paper introduces a novel aspect-level sentiment analysis model that combines adversarial training with dependency parsing. The model leverages BERT for word vector conversion and employs an adjacency matrix to capture syntactic dependencies. Multi-head attention combines these features, while adversarial training enhances robustness. This approach enables accurate sentiment polarity predictions at the aspect level.

The primary contributions of this paper can be summarized as follows:

1. The introduction of dependency parsing information in aspect-level sentiment analysis. By constructing an adjacency matrix of syntactic dependency relations, the model can more precisely capture the semantic correlations between different aspects in the text, thereby improving the precision and accuracy of sentiment analysis;
2. To better integrate the features of both BERT and syntactic dependency relations, a multi-head attention mechanism is adopted. This mechanism considers different feature word vectors simultaneously, allowing the model to comprehend the semantic information of the text more comprehensively, thereby enhancing the performance;
3. In order to bolster the robustness and generalizability of the model, an adversarial training mechanism is introduced. By applying small perturbations to the BERT embedding layer, FGM (fast gradient method) can make the model better resist attacks from adversarial samples, thus improving the model's stability and reliability in real-world applications.

2. Related Work

2.1. Aspect-Level Sentiment Analysis

Aspect-level sentiment analysis is a vital task within sentiment analysis, concentrating on the sentiment polarity of particular aspect terms within a sentence. Traditional sentiment analysis methods often target entire documents or single sentences, whereas aspect-based sentiment analysis pays closer attention to more refined sentiment evaluations of specific entities. In past research, the use of traditional machine learning methods for sentiment classification has been a common practice. For instance, Kiritchenko et al. used SVM to detect aspect terms and sentiment in customer reviews [9]; Akhtar et al. employed SVM and CRF for Hindi sentiment classification with good results [10]; Patra et al. used CRF for aspect-level sentiment classification in the domains of Laptop and Restaurant datasets, providing valuable references for consumers and manufacturers [11]. However, these methods require manual feature selection and semantic information extraction, which can

reduce the error of opinion word matching but still have limitations. For example, feature extraction from dataset texts requires a significant amount of labor, and the final sentiment analysis results are highly dependent on feature quality, but are incapable of modeling the dependencies between the provided aspect terms and their surrounding contexts.

Comparatively, deep neural networks possess more intricate model architectures and stronger feature extraction capabilities, eliminating the necessity for manual feature extraction, reducing labor costs. With the improvements in computer hardware performance and the widespread use of the Internet, deep neural networks are no longer limited by hardware computing power and data samples. In the realm of sequence models with a focus on attention, researchers have proposed a variety of methodologies. For example, Cheng et al. improved the feature extraction capacity of the Transformer bidirectional encoder through an extended context module and proposed a component focusing module to address the issue of average pooling [12]. Huang et al. proposed the AGSNP model, which combined attention mechanisms and achieved good results [13]. Ayetiran proposed a CNN and BiLSTM variant that combined high-level semantic feature extraction and sentiment polarity prediction [14]. In models focusing on syntactic information, Zeng et al. utilized affective knowledge to enhance word representations, forming a heterogeneous graph based on dependency trees, and designed a multi-level Semantic-HGCN to encode the graph for sentiment prediction [15]. Gu et al. proposed the EK-GCN model, which uses an external sentiment dictionary to assign sentiment scores to individual words within a sentence, constructing an emotional matrix to partially compensate for the shortcomings of the syntactic dependency tree [16]. In models focusing on contextual modeling, Xiao et al. proposed a novel GNN-based deep learning model, leveraging a POS-guided syntactic dependency graph for RGAT to eliminate noise and designing a syntactic distance attention-guided layer for DCGCN to extract semantic dependencies between contextual words [17]. Mewada et al. utilized affective knowledge to enhance word representations, forming a heterogeneous graph based on dependency trees, and then designing a multi-level Semantic-HGCN to encode the graph for sentiment prediction [18]. Xu et al. proposed a sentiment analysis model based on dynamic local context and dependency clusters, which dynamically captured the scope of local context and extracted semantic information, achieving good results [19]. Mao et al. proposed a multi-task learning approach, incorporating a novel gated bridging mechanism (GBM), which achieved superior performance in aspect-based sentiment analysis by effectively filtering irrelevant information and dynamically extracting features for each subtask using a weighted-sum pooling strategy [20].

2.2. Dependency Analysis

Dependency parsing, also known as dependency syntax analysis [21], aims to identify the interdependent relationships between words in a given text and find the corresponding dependent words (tail nodes) for each word (head node), which facilitates a deeper comprehension of the entire sentence's meaning. This is also one of the more critical technologies in the field of NLP. The representation is through directed arrows from the central word to its dependent words, forming directed graphs. Dependency projection trees, and dependency trees are common ways to express dependency structures. Taking the sentence "The iced Americano at this airport tastes good" as an example, the expression of its dependency tree is as follows (Figure 1):

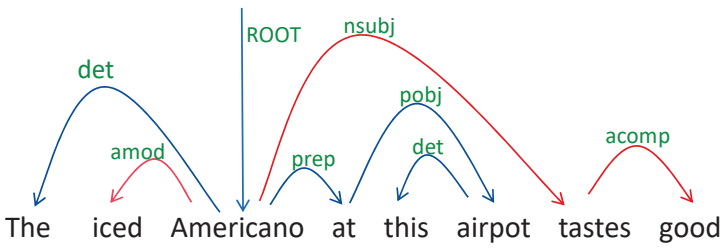


Figure 1. Example diagram of dependency syntax tree.

Dependency syntax analysis is typically represented as a tree structure, where the nodes of the tree represent words, the edges represent the dependency relationships between words, and the parent node of the tree indicates the governor. Some commonly used dependency relation labels, and their meanings in dependency syntax analysis, are presented in Table 1.

Table 1. Partial dependency relationship labels and their meanings.

Labels	Meanings
ROOT	Root node
det	Dependency
amod	Adjectives
nsubj	Noun subjects
prep	Prepositional modifiers
pobj	Object of a preposition
acomp	Complement of an adjective

2.3. Adversarial Training

In the domain of computer vision (CV), it is essential to enhance the robustness of models through adversarial attacks and defenses. For instance, in autonomous driving systems, it is crucial to prevent models from misclassifying red lights as green due to random noise. Similarly, in natural language processing (NLP), adversarial training exists, primarily as a regularization technique aimed at enhancing model generalization.

In 2014, Szegedy et al. introduced the concept of adversarial examples, which is considered a pioneering work in the field [22]. For models processing text input data, the added perturbations can be categorized into two types: discrete, where perturbations are directly applied to the text; and continuous, where tiny perturbations are introduced into the word vector matrix. This paper employs the latter approach for adversarial training. Current popular adversarial training methods include the fast gradient sign method (FGSM) [23], fast gradient method (FGM), projected gradient descent (PGD) [24], free adversarial training (FreeAT) [25], and free large-batch (FreeLB) [26].

The core of adversarial training lies in constructing perturbations that enable the model to recognize diverse adversarial examples. Adversarial training algorithms first generate perturbations using adversarial attacks, then combine these perturbations with original samples to create adversarial examples. Subsequently, the model parameters are adjusted via backpropagation to minimize the loss function. This process can be defined as a max–min optimization problem, where the maximization problem involves finding perturbations that maximize the loss function for generating adversarial examples, while the minimization problem involves minimizing the loss function and updating model parameters, thereby endowing the model with robustness to adapt to such perturbations. Adversarial training can be uniformly represented as a min-max formula, as shown in the following equation:

$$min_{\theta}\mathbb{E}_{(x,y)\sim\mathcal{D}}[max_{\Delta x\in\Omega}L(x+\Delta x,y;\theta)] \quad (1)$$

where $\mathcal{D}$ represents the dataset, x represents the inputs, y represents the labels, and θ is the model parameter that represents the parameter vector of the neural network, $L(x+\Delta x,y;\theta)$ is a single sample of *loss*, Δx is the perturbation and Ω is the perturbation space. Then after the neural network function, the loss obtained by comparing with the label y, $max_{\Delta x\in\Omega}L()$ denotes the optimization objective.

2.4. Attention Mechanisms

In 2014, the Google Mind team's paper brought attention mechanisms into the spotlight [27]. Initially introduced for image processing tasks, attention mechanisms have proven to be effective in other fields as well. Experimental validations have demonstrated the theoretical feasibility of attention mechanisms, and empirical results in the field of

NLP have shown their efficacy in sentiment analysis tasks, highlighting their significant research value. This method is capable of effectively extracting key features, and as such, it is currently widely employed to enhance the performance of sentiment analysis models. The attention mechanism simulates the cognitive process of the human brain, quickly extracting valuable information from extensive text data and assigning higher weights to important information while assigning lower weights to other information.

Bahdanau et al. were the first to introduce attention mechanisms into machine translation based on the encoder–decoder model, successfully translating long sentences [28]. Despite potential issues with the encoding quality, attention mechanisms addressed this by allocating distinct weights to words in the encoding module based on their importance, leading to notable experimental results. The introduction of attention mechanisms has solved the problem of poor coding module quality for machine translation of long sentences, and this technology has been widely applied in the field of NLP, playing an especially important role in sentiment analysis tasks.

The unified computation method of the attention mechanism can be represented as follows:

$$Attention(Q, K, V) = softmax\left(QK^T\right)V \tag{2}$$

In attention mechanisms, Q represents the query vector, K denotes the key vectors within a sentence, typically used for relevance calculations, and V represents the value vectors. Attention weights are obtained through a normalization method, which fundamentally maps the query vector to a series of relationships among key-value pairs. The structure can be visualized as follows (Figure 2):

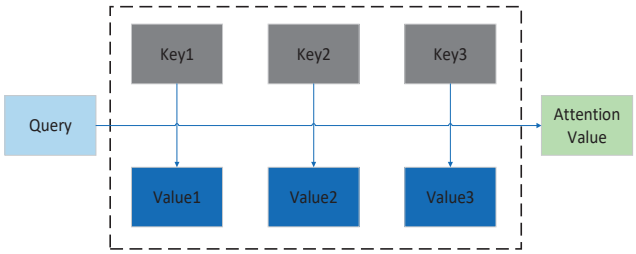


Figure 2. Attention mechanism structure.

3. Overall Model Design

This section begins with a description of the aspect-level sentiment analysis task, followed by the model structure.

3.1. Task Definition

The model input is the given text $W = \{w_1, w_2, \ldots a, \ldots o, \ldots w_n\}$, where a denotes an aspect word and o denotes an opinion word, and the model outputs the sentiment polarity $y \in \{positive, negative, neutral\}$ corresponding to the aspect. Our model leverages the pre-trained language model BERT to generate and train word vectors.

3.2. Model Architecture

The model discussed in the paper comprises the following six main components: a text embedding layer, BERT encoding layer, syntactic dependency relation information layer, adversarial training layer, multi-head attention layer, and an output layer. The model's overall structure is depicted in Figure 3.

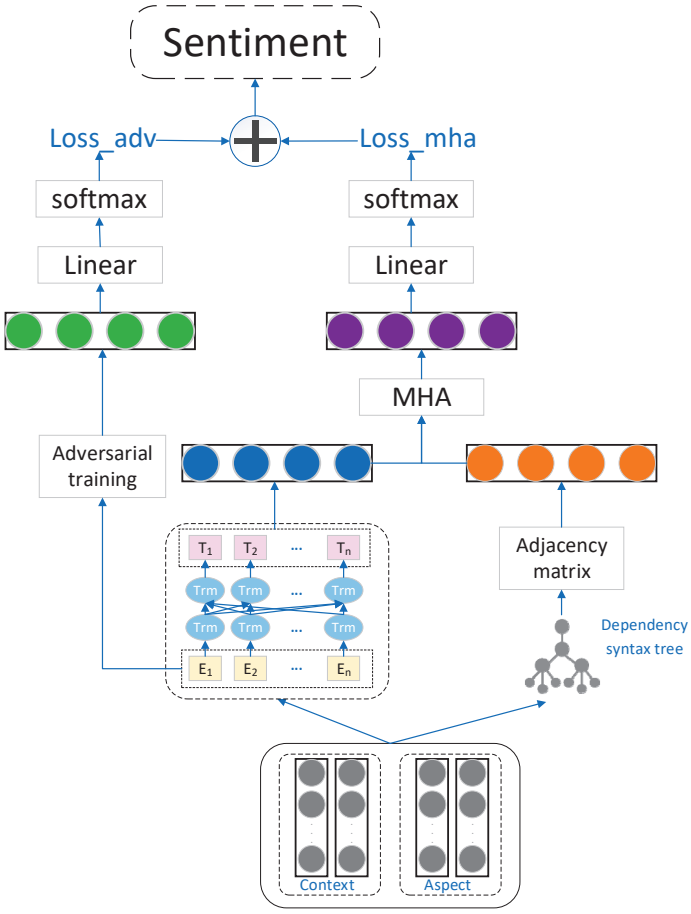


Figure 3. Model structure diagram.

3.2.1. Text Embedding Layer

For a sentence $W = \{w_1, w_2, \ldots a, \ldots o, \ldots w_n\}$, use the pre-training model BERT to map each word onto an embedding vector $e_i \in R^{d \times 1}$, where d represents the dimension of the word vector:

$$W_{Bert} = Bert(W) \tag{3}$$

To fully leverage the power of BERT in model training, the text is formatted into the structure of "[CLS] + context + [SEP] + target + [SEP]". In this format, "[CLS]" and "[SEP]" are special token markers utilized by BERT. "[CLS]" serves as a unique classification token marker that encapsulates classification-related information, while "[SEP]" functions as a separator to demarcate distinct sequences when multiple sequences are input. By adhering to the formatting requirements specified by BERT for text classification tasks, the effectiveness of BERT is maximized.

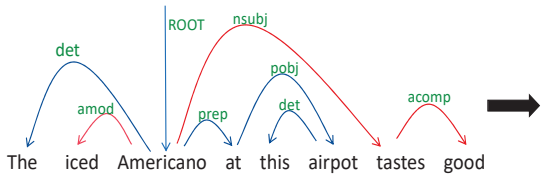
3.2.2. BERT Encoding Layer

The BERT encoder is constructed using Transformer blocks from the Transformer model [29]. For BERT-BASE, these blocks are employed in 12 layers, each consisting of 12 multi-head attention blocks. After passing through the BERT model, the output is a new sequence with the same length as W_{Bert}, represented as $H_{Bert} = \{h_{CLS}, h_1, \ldots, h_{n-1}, h_{SEP}, h_a, h_{SEP}\}$ as the representation of hidden vectors. Here, "h_{CLS}" is the hidden vector for the classification token, "h_1" to "h_{n-1}" are the hidden vectors for the context tokens, "h_{SEP}" represents the hidden vectors for the separator tokens, and "h_a" represents the hidden vectors for the aspect words.

3.2.3. Dependency Syntax Relation Information Layer

The text is simultaneously processed to establish syntactic dependency relations. In this paper, the StanfordCoreNLP tool is used to obtain the syntactic dependency tree of the text [30]. This is done by capturing the grammatical structure of sentences to extract dependency analysis; the output is a list containing multiple tuples. For example, in the sentence "The iced Americano at this airport tastes good", the output is [('ROOT',0,3), ('det',3,1), ('amod',3,2), ('nsubj',7,3), ('prep',4,3), ('pobj',6,4), ('det',6,5), ('acomp',8,7)]. In this sentence, there are a total of eight elements, so ('amod',3,2) indicates an adjective, where "Americano" depends on "iced". Words in the sentence are encoded starting from 1 to the end of the sentence. The numbers in the tuple represent the positions of the words, and the numbers before and after represent the dependency relationship, where the first number is the head and the second number is the child, indicating that the latter depends on the former. Then, the dependencies are mapped onto a directed graph. The syntactic dependency tree can be conceptualized as graph G with n nodes, where the nodes correspond to the words in the sentence, and the edges represent the syntactic dependencies between words. The dependency parse tree of a sentence is represented as $G = \{V, A\}$, where V stands for all the nodes, which are the words $\{w_1, w_2, \ldots a, \ldots o, \ldots w_n\}$; and $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix, where $A_{ij} = 1$ if there is a syntactic dependency between word w_i and word w_j, and $A_{ij} = 0$ otherwise. Each word in the sentence is adjacent to itself, which implies setting all diagonal elements of the adjacency matrix to 1 [31].

Here is how the syntactic dependency tree and its transformed adjacency matrix are depicted (Figure 4):



	The	iced	Americano	at	this	airpot	tastes	good
The	1	0	1	0	0	0	0	0
iced	0	1	1	0	0	0	0	0
Americano	1	1	1	1	0	0	1	0
at	0	0	1	1	0	1	0	0
this	0	0	0	0	1	1	0	0
airpot	0	0	0	1	1	1	0	0
tastes	0	0	1	0	0	0	1	1
good	0	0	0	0	0	0	1	1

Figure 4. Syntactic dependency and adjacency matrix.

Next, the adjacency matrix is expanded into a one-dimensional vector and connected to the elements in the matrix row by row or column by column. The unfolded vector is used as an input for the next step of model processing. This converts the information of the adjacency matrix into a vector V_{adj}.

3.2.4. Adversarial Training Layer

The model uses the FGM (fast gradient method) for adversarial training on the BERT embedding layer vectors. FGM stands out from other methods due to its simplicity, ease of use, and computational efficiency. It generates adversarial samples with minimal parameter updates, making it practical for real-world applications with low computational costs, especially with large datasets and complex models. Despite potential variations in performance, FGM typically enhances model robustness against common adversarial attacks. Thus, FGM is a practical choice, particularly in resource-constrained scenarios or where rapid implementation is crucial. By performing gradient ascent based on the specific gradients, it aims to obtain better adversarial samples without significantly altering the distribution of the original samples, thereby allowing the model to adapt to such perturbations. Assuming that the embedding layer vectors $V = \{v_1, v_2, \ldots, v_n\}$ of the input text sequence are x, the perturbation on the embedding layer is as follows:

$$\Delta x = \epsilon \frac{g}{\| g \|_2} \tag{4}$$

$$g = \nabla_x L(x, y; \theta) \tag{5}$$

$$V_{adv} = V + \Delta x \tag{6}$$

After the adversarial training, the obtained feature vectors are denoted as V_{adv}.

3.2.5. Multi-Head Attention Mechanism Layer

After flattening the hidden features H_{Bert} obtained from BERT's output, we obtain H'_{Bert}. Then, we concatenate it with the feature vector obtained after adversarial training to obtain the new hidden feature $Z = \left[H'_{Bert}, V_{adj}\right]$. Z represents the input to the multi-head attention module. By utilizing three different weights W_q, W_k, W_v in the attention layer, we can calculate the resulting vector q, k, v. The steps of the multi-head attention mechanism involve linearly transforming the query (Q), key (K), and value (V) through parameter matrices. Then, scaled dot-product operations are performed multiple times before concatenating the results. First, the score for each input feature is calculated: $score = k \times q$. Then, each score is normalized by dividing it by the square root of the dimension of the weight matrix $\sqrt{d_k}$. Next, the softmax function is applied to the normalized scores. Finally, the softmax result is multiplied by the value V. The formula is as follows:

$$a = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

The multi-head attention mechanism assigns weighted attention scores to each word in the sentence using multiple attention mechanisms. By increasing the weight coefficients of important information, the model focuses more on words crucial for sentiment analysis, thus further enhancing the accuracy of sentiment analysis. The multi-head attention mechanism consists of multiple heads, each capable of generating different attention distributions, thereby addressing long-range dependencies. Built upon the attention mechanism, the multi-head attention mechanism significantly outperforms standard attention mechanisms, allowing for parallel processing of information in different positional and representational subspaces. With each set of attention projected into different spaces, and considering m as the number of attention heads, the calculation formula is as follows:

$$head_i = a_i \tag{8}$$

$$e = MultiHead(Q, K, V) = Concat(head_1, head_2, \ldots, head_m)W^O \tag{9}$$

Wherein, W^O represents the weight vector, which can be learned through the training process. The Concat function indicates the concatenation of the vectors after the attention computation, and $head_i$ represents the i-th attention mechanism.

Finally, all encoding vectors are weighted and summed to obtain a comprehensive hidden expression e.

3.2.6. Output Layer

Considering that the adversarial perturbations in adversarial training are relatively small values, to prevent the word vectors from becoming too large, which could cause the tiny perturbations to lose their effectiveness, it is necessary to normalize the word vectors. Normalization ensures that the values of the word vectors remain within a reasonable range, allowing the model to be sensitive to the small adversarial perturbations. It is described as follows:

$$V'_{\text{adv}} = \frac{V_{adv} - E(V_{adv})}{\sqrt{Var(V_{adv})}} \tag{10}$$

$$E(V_{adv}) = \sum_{i=1}^{K} f_i V_{advi} \tag{11}$$

$$Var(V_{adv}) = \sum_{i=1}^{K} f_i (V_{advi} - E(V_{adv}))^2 \tag{12}$$

where V_{adv} denotes the original word vector V'_{adv} denotes the normalized word vector and f_i denotes the frequency of the ith word in the training sample.

The fused features and adversarial features of the multi-head attention mechanism are, respectively, used as inputs to the *Softmax* classifier, after which the fused features of the multi-head attention mechanism and the real labels can be calculated as the classification loss $Loss_{mha}$, which is calculated by the following formula:

$$Loss_{mha} = \sum_{i=1}^{N} \left\{ y_i log\hat{y}_i + (1 - y_i) log\left(1 - \hat{y}_i\right) \right\} \tag{13}$$

Wherein y_i represents the true category, $\hat{y}_i$ represents the predicted category, and N is the overall number of samples.

Subsequently, the adversarial features and the true labels are used as inputs to the classifier for calculating the adversarial training loss $Loss_{adv}$, with the following formula:

$$Loss_{adv} = -\frac{1}{N} \sum_{n=1}^{N} logp(y_n \mid x + \Delta x, \theta) \tag{14}$$

In this loss function, the variable is $x + \Delta x$, where Δx represents the adversarial perturbation, N is the overall number of samples, y_n is the corresponding label, and θ is the model's parameters that represents the parameter vector of the neural network. Therefore, the actual loss of the model is as follows:

$$Loss = Loss_{mha} + Loss_{adv} \tag{15}$$

Furthermore, the gradients of $Loss_{mha}$ and $Loss_{adv}$ with respect to the model parameters are computed first. Subsequently, these gradients, along with a predefined learning rate, are utilized to update the model parameters, aiming to progressively decrease the overall loss. Until it satisfies the predetermined maximum number of iterations, this iterative process continues. The generated adversarial training samples are used together with the original samples for model training. This approach can expand the dataset size and effectively enhance the model's generalization performance and classification accuracy.

4. Experimental Analysis

4.1. Experimental Dataset and Experimental Environment

The model in this paper was mainly evaluated on the SemEval2014 Task4 public dataset, which consists of reviews from two domains, Laptops and Restaurants [32]; these datasets are partitioned into a training set and a test set. The aspect words and their corresponding sentiment polarity in the dataset have been labelled, where -1 represents negative, 0 represents neutral and 1 represents positive. The dataset's fundamental statistics are provided in Table 2.

Table 2. Basic statistical information of the dataset.

Datasets	Negative		Neutral		Postive	
	Train	Test	Train	Test	Train	Test
Laptops	851	128	455	167	976	337
Restaurants	807	196	637	196	2164	727

Table 3 illustrates the pertinent configuration of the experimental environment in this paper.

Table 3. Configuration of experimental environment.

Experimental Environment Configuration Table	Configuration Information
Operating System CPU	AMD Ryzen 7 7735H with Radeon Graphics 3.20 GHz
Graphics card	NVIDIA GeForce RTX 4060
Deep Learning Framework	Pytorch
Development Environment	Pycharm

4.2. Experimental Parameter Setting

The experiment used the pretrained language model BERT to generate word vectors. The generated word vectors have a dimension of 768, with a hidden-layer dimension of 300. The dropout rate is set to 0.1, and the learning rate is 2×10^{-5}. The batch size for each input data is 32, and the optimizer used is Adam [33].

4.3. Evaluation Indicators

In the experiment, the evaluation metrics used were *Accuracy* and *Macro-averaged F*1 score [34,35]. Accuracy denotes the proportion of correctly classified positive and negative samples to the total number of samples. The *F*1 score is the harmonic mean of precision and recall, encompassing both precision and recall in the evaluation of the model. The macro-averaged *F*1 score is the average of the *F*1 scores for each category, which helps to avoid the issue of artificially high accuracy due to imbalanced data. The specific formulas are as follows:

$$Accurary = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FP} \tag{18}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{19}$$

$$MF1 = \frac{1}{C} \sum_{k=1}^{c} F1_k \tag{20}$$

wherein *TP* represents the number of positive samples correctly predicted as positive, *FN* denotes the number of positive samples mistakenly predicted as negative, *FP* indicates the number of negative samples erroneously predicted as positive, and *TN* signifies the number of negative samples accurately predicted as negative. Precision and Recall denote the precision rate and recall rate, respectively, while *C* represents the number of sentiment categories.

To evaluate the significance of the improved results, we also added kappa consistency as a statistical test indicator. The Kappa coefficient, which is a statistical measure of consistency ranging between 0 and 1, is elaborated upon in Table 4. A larger coefficient signifies greater precision in data classification. Its calculation formula is as follows:

$$K = \frac{P_o - P_e}{1 - P_e} \tag{21}$$

Table 4. Kappa coefficient table.

Coefficient	0.8–1.0	0.6–0.8	0.4–0.6	0.2–0.4	0–0.2
Level	Almost perfect	Substantial	Moderate	Fair	Slight

P_o represents the overall classification accuracy. The calculation formula for P_e is as follows:

$$P_e = \frac{a_1 \times b_1 + a_2 \times b_2 + \cdots + a_k \times b_k}{n \times n} \tag{22}$$

a_k represents the actual sample size of class k, b_k represents the predicted sample size of class k, and the total sample size is n.

4.4. Comparative Experiments

The paper selected seven representative aspect-level sentiment analysis models to compare with the model provided in this paper, and their descriptions are as follows:

(1) LSTM [36] is an aspect-level sentiment analysis model based on long short-term memory networks that uses a recurrent neural network structure for modeling and can capture temporal information in text. It performs sentiment classification by integrating the target word and context relationships through two LSTM layers that depend on the target;
(2) TD-LSTM [37] utilizes LSTM to encode the contexts on both sides of the aspect term from different directions, and performs sentiment classification by concatenating the resulting feature representations;
(3) MemNet [38] is a deep memory network model combined with an attention mechanism. By constructing multiple computational layers, each input layer adaptively selects deeper-level information and captures the correlation between each context word and the aspect via attention layers. The output of the final attention layer is utilized for sentiment polarity assessment;
(4) IAN [39] utilizes two LSTM layers to acquire the hidden representations of the context and aspect terms. To precisely capture the semantic relationship between context words and the aspect term, an interactive attention mechanism is incorporated;
(5) RAM [40] is a memory neural network model based on a recurrent attention mechanism that can effectively obtain the sentiment features between words that are farther apart;
(6) AEN [41] utilizes an encoder with an attention mechanism to establish a sentiment analysis model between the context and its corresponding aspect term;
(7) ASGCN [42] constructs a graph convolutional network on the sentence's dependency tree to extract syntactic information. By integrating attention with masked aspect vectors and semantic information, it enhances sentiment classification performance;
(8) GPT3+Prompt [43] is a language model that can be guided to perform aspect-level sentiment analysis tasks and generate relevant text by adding prompts.

Among all comparison models, the accuracy of the ASGCN model reached 75.55% and 80.77% on both datasets, respectively. This is because the ASGCN model constructs a graph convolutional network on the dependency tree of sentences, utilizing syntactic information to extract semantic relationships and improving the accuracy of sentiment classification.

The accuracy rates of LSTM and TD-LSTM on the two datasets reached 66.77% and 74.29%; and 67.71%, 75.36%, respectively. TD-LSTM improved the LSTM model, but because LSTM cannot reflect the interaction information between aspect words and text sentences, and LSTM processes sentences in the order of text sequences, the semantic information learned is not comprehensive enough, and too-long sentences can cause slow the gradient descent. The MemNet model's attention mechanism for selecting deeper-level information may falter in filtering out noisy context words, potentially leading to reduced classification performance. Despite the IAN model's precise capture of semantic relationships, it may face challenges with highly ambiguous context terms. The AEN model's focus on context information might overlook subtle sentiment nuances. Lastly, the RAM model's recurrent attention mechanism may introduce computational complexity and training instability. The above reasons have led to the poor performance of these models.

As indicated in Table 5, the BAMD model surpasses other models in both Accuracy and Macro-F1 scores. On the two datasets, the accuracy rates of the BAMD model reached 76.02% and 83.04%, respectively. Our model offers several advantages over baseline models: first, by integrating dependency parsing information, we accurately capture semantic correlations between different aspects in the text. Second, employing a multi-head attention mechanism enables a comprehensive understanding of semantic information within the text. Lastly, the introduction of adversarial training enhances the model's stability and reliability in real-world applications.

Table 5. Comparing the experimental results of the model on two publicly available datasets.

Comparative Models	Laptops			Restaurants		
	Accurary	**Macro-F1**	**Kappa**	**Accurary**	**Macro-F1**	**Kappa**
LSTM	66.77	61.78	-	74.29	62.58	-
TD-LSTM	68.81	64.67	-	76.00	64.51	-
MemNet	70.64	65.17	-	79.61	69.64	-
IAN	71.20	66.69	-	76.86	66.71	-
RAM	72.32	67.90	0.6745	76.92	68.71	0.7148
AEN	73.69	68.59	0.6886	77.06	69.35	0.7262
ASGCN	75.55	71.05	0.6904	80.77	72.02	0.7377
GPT3 + Prompt	**77.87**	**73.04**	-	**85.45**	**78.96**	-
BAMD(Ours)	76.02	71.54	**0.7171**	83.04	76.61	**0.7853**

Although our model is only 1 to 2.5 percentage points less effective than the closed-source GPT3+Prompt, we acknowledge this difference. Our research suggests that, while our model may be lightweight, with fewer parameters and a smaller memory footprint, this lightweight nature makes it more feasible for deployment and operation in resource-constrained environments, with lower computational costs. While our model may slightly lag behind larger models in performance, its lightweight characteristics provide greater flexibility and feasibility for specific applications in certain scenarios. We will continue to strive for improvement and look forward to achieving better results in future research.

Moreover, it can be clearly seen from the chart that our model's Kappa value is significantly better than the compared models. This indicates that our model can still maintain high classification consistency while considering randomness. The significance of this improvement is not only reflected in the Kappa value, but also in the robustness and generalization ability of the model on different datasets. Therefore, our model performs more reliably and stably in solving this classification task. The optimal performance metrics of each model on two datasets have been bolded in the table.

4.5. Ablation Experiment

To verify the importance of the three major modules designed in this paper, a series of ablation experiments were conducted.

For each ablation experiment, we can infer the importance of each component to model performance by the degree of degradation in the evaluation metrics:

The ablation experiment without Adversarial Training (w/o AT) exhibited a decrease in performance when compared to the original model. This is because adversarial training plays a crucial role in enhancing model robustness and generalization capabilities. Without adversarial training, the model is more susceptible to the influence of biased or noisy samples, leading to a decrease in performance. Therefore, adversarial training is vital for improving the robustness of the model.

Our model significantly outperformed the version without multi-head attention (w/o MHA). The multi-head attention mechanism aids in better integrating features from BERT and syntactic dependency relations, enhancing the model's attention to different aspects of the text and its representational power. If the multi-head attention mechanism is removed, the model may not effectively capture sentiment information across different aspects,

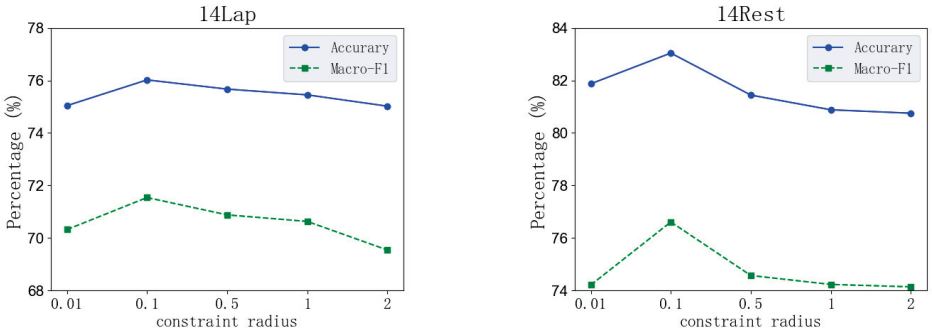
resulting in a decline in performance. It is evident that multi-head attention is important for enhancing the model's representational capabilities.

The absence of syntactic dependency relations resulted in varying degrees of decline in both Accuracy and Macro-F1 scores. Syntactic dependency relations provide structural information between words in the text, which helps the model to better understand the semantic and logical relationships within sentences. If syntactic dependency relations are removed, the model may not effectively utilize the structural information of the sentence, leading to a decrease in performance. Therefore, syntactic dependency relations are important for enhancing the model's semantic comprehension.

In summary, adversarial training, multi-head attention mechanism, and syntactic dependency relations each play a significant role in improving model performance. Together, they constitute the key components of the model proposed in this paper.

4.6. Analysis of Model Parameters

To investigate the impact of the constraint radius of the perturbation constraint space S, i.e., the value of ε, on model performance in adversarial training, this paper set ε values to 0.01, 0.1, 0.5, 1, and 2. The accuracy and MF1 scores were tested on both the 14Lap and 14Rest datasets (Figure 5). The experimental results, as shown in the figure below, indicate that introducing adversarial samples during the training stage can enhance the model's resilience to attacks. The model performs optimally when the ε value is 0.1; however, when the ε value is too large, both the model's accuracy and MF1 scores exhibit a downward trend. This phenomenon may be due to the larger perturbation values added, which resulted in significant differences between the generated adversarial samples and the original samples. Although they shared the same label, the model's accuracy in identifying these adversarial samples decreased, subsequently leading to a decline in model performance.



(**a**) (**b**)

Figure 5. (**a**) The accuracy of the model under different constraint radii (14Lap); and (**b**) the accuracy of the model under different constraint radii (14Rest).

4.7. Case Study

In order to reflect the effectiveness of the proposed approach, several specific examples were analyzed. Based on Table 6, we extracted the classification results of some typical examples for comparative analysis (Table 7).

Table 6. Results of ablation experiment.

Models	Laptops		Restaurants	
	Accurary	**Macro-F1**	**Accurary**	**Macro-F1**
w/o DS	74.52	70.31	80.57	76.22
w/o AT	73.45	69.63	78.63	73.25
w/o MHA	73.58	69.97	79.78	74.56
BAMD	76.02	71.54	83.04	80.26

Table 7. Typical data experiment examples.

Num	Examples	TD-LSTM	ASGCN	BAMD	Label
1	The *food* is great but the service was dreadful!	Negative (×)	Positive (√)	Positive (√)	Positive
2	I'm delighted to return to the familiar embrace of *Apple's operating system*.	Positive (√)	Negative (×)	Positive (√)	Positive
3	Did not enjoy the new *Windows 8* and touchscreen functions.	Natural (×)	Positive (×)	Negative (√)	Negative

"√" in the table represents the model's correct judgment of emotional polarity, while "×" represents the model's incorrect judgment of emotional polarity.

For the first example sentence, due to the existence of two aspect terms, namely "food" and "service", TD-LSTM focused on the opinion word "dreadful" related to "service", considering it as the opinion word for the aspect term "food", leading to an incorrect matching between aspect terms and opinion words, resulting in a negative sentiment judgment. In the second example sentence, the syntactic distance between "Apple's operating system" and its opinion word "delighted" is too great. The aspect sentiment graph convolutional network (ASGCN) model failed to capture the relationship between them based on syntactic information, which resulted in an incorrect sentiment polarity judgment. The third example also contains two aspect terms, where TD-LSTM failed to accurately match aspect terms with opinion words, and ASGCN failed to capture the feature representation of the negation word "did not". In contrast, BAMD combines both adversarial training and dependency syntax information, and thus can make accurate judgments.

5. Conclusions

This paper introduces an aspect-level sentiment analysis model that leverages adversarial training in conjunction with dependency syntax parsing. By employing BERT for word vector transformation, integrating feature extraction from syntactic dependency relations, and utilizing multi-head attention mechanisms along with adversarial training techniques, the proposed model is capable of predicting the sentiment polarity of specific aspects within sentences. On two public aspect-level sentiment analysis datasets, our model achieves higher accuracy and MF1 scores compared to the baseline models, validating the effectiveness of our approach. However, the model presented in this paper has certain limitations. For instance, the generated dependency syntax relations may contain data noise, and the influence of part-of-speech tags and other syntactic information on the task is not considered. The choice of the adversarial training method can be adjusted to optimize model performance for specific datasets. Future work will focus on further improving and enhancing the model to address these challenges. Specifically, we will explore methods to reduce data noise in generated dependency relations, incorporate part-of-speech tags and other syntactic information, and optimize adversarial training methods for specific datasets. These advancements aim to enhance the model's performance and applicability in aspect-level sentiment analysis, thereby promoting its development and application in various domains.

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

Funding: This research was supported by the National Key Research and Development Program of China (2022YFC3302300), Advanced Research Project (7090201050307) and National 242 Information Security Program (2023A105).

Data Availability Statement: The data presented in this study can be provided upon request.

Acknowledgments: The author thanks my supervisors and colleagues for their help, which enabled me to complete this article.

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

References

1. Saberi, B.; Saad, S. Sentiment analysis or opinion mining: A review. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2017**, *7*, 166–1666.
2. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. [CrossRef]
3. Thet, T.T.; Na, J.C.; Khoo, C.S.G. Aspect-based sentiment analysis of movie reviews on discussion boards. *J. Inf. Sci.* **2010**, *36*, 823–848. [CrossRef]
4. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
6. Zhuang, L.; Wayne, L.; Ya, S.; Jun, Z. A robustly optimized BERT pre-training approach with post-training. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, Huhhot, China, 13–15 August 2021; pp. 1218–1227.
7. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5753–5763.
8. Mao, R.; Liu, Q.; He, K.; Li, W.; Cambria, E. The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. *IEEE Trans. Affect. Comput.* **2022**, *14*, 1743–1753. [CrossRef]
9. Kiritchenko, S.; Zhu, X.; Cherry, C.; Mohammad, S. Detecting aspects and sentiment in customer reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval), Dublin, Ireland, 23–24 August 2014; pp. 437–442.
10. Akhtar, M.S.; Ekbal, A.; Bhattacharyya, P. Aspect based sentiment analysis in Hindi: Resource creation and evaluation. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 2703–2709.
11. Patra, B.G.; Mandal, S.; Das, D.; Bandyopadhyay, S. Ju_cse: A conditional random field (crf) based approach to aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 370–374.
12. Cheng, L.C.; Chen, Y.L.; Liao, Y.Y. Aspect-based sentiment analysis with component focusing multi-head co-attention networks. *Neurocomputing* **2022**, *489*, 9–17. [CrossRef]
13. Huang, Y.; Peng, H.; Liu, Q.; Yang, Q.; Wang, J.; Orellana-Martín, D.; Pérez-Jiménez, M.J. Attention-enabled gated spiking neural P model for aspect-level sentiment classification. *Neural Netw.* **2023**, *157*, 437–443. [CrossRef]
14. Ayetiran, E.F. Attention-based aspect sentiment classification using enhanced learning through CNN-BiLSTM networks. *Knowl.-Based Syst.* **2022**, *252*, 109409. [CrossRef]
15. Zeng, Y.; Li, Z.; Chen, Z.; Ma, H. Aspect-level sentiment analysis based on semantic heterogeneous graph convolutional network. *Front. Comput. Sci.* **2023**, *17*, 176340. [CrossRef]
16. Gu, T.; Zhao, H.; He, Z.; Li, M.; Ying, D. Integrating external knowledge into aspect-based sentiment analysis using graph neural network. *Knowl. Based Syst.* **2023**, *259*, 110025. [CrossRef]
17. Xiao, L.; Xue, Y.; Wang, H.; Hu, X.; Gu, D.; Zhu, Y. Exploring fine-grained syntactic information for aspect-based sentiment classification with dual graph neural networks. *Neurocomputing* **2022**, *471*, 48–59. [CrossRef]
18. Mewada, A.; Dewang, R.K. SA-ASBA: A hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language BERT model with extreme gradient boosting. *J. Supercomput.* **2023**, *79*, 5516–5551. [CrossRef]
19. Xu, M.; Zeng, B.; Yang, H.; Chi, J.; Chen, J.; Liu, H. Combining dynamic local context focus and dependency cluster attention for aspect-level sentiment classification. *Neurocomputing* **2022**, *478*, 49–69. [CrossRef]
20. Mao, R.; Li, X. Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 13534–13542.
21. Nguyen, D.Q.; Verspoor, K. An improved neural network model for joint POS tagging and dependency parsing. *arXiv* **2018**, arXiv:1807.03955.
22. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
23. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
24. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
25. Shafahi, A.; Najibi, M.; Ghiasi, M.A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L.S.; Taylor, G.; Goldstein, T. Adversarial training for free! *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 3358–3369.
26. Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; Liu, J. Freelb: Enhanced adversarial training for natural language understanding. *arXiv* **2019**, arXiv:arxiv:1909.11764.
27. Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2204–2212.

28. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
30. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
31. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
32. Kirange, D.K.; Deshmukh, R.R. Emotion classification of restaurant and laptop review dataset: Semeval 2014 task 4. *Int. J. Comput. Appl.* **2015**, *113*, 17–20.
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
34. Chinchor, N.; Sundheim, B.M. MUC-5 evaluation metrics. In Proceedings of the Fifth Message Understanding Conference (MUC-5), Baltimore, MD, USA, 25–27 August 1993.
35. Yang, Y.; Liu, X. A re-examination of text categorization methods. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, 15–19 August 1999; pp. 42–49.
36. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [CrossRef]
37. Tang, D.; Qin, B.; Feng, X.; Liu, T. Effective LSTMs for target-dependent sentiment classification. *arXiv* **2015**, arXiv:1512.01100.
38. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. *arXiv* **2016**, arXiv:1605.08900.
39. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. *arXiv* **2017**, arXiv:1709.00893.
40. Chen, P.; Sun, Z.; Bing, L.; Yang, W. Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 452–461.
41. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Attentional encoder network for targeted sentiment classification. *arXiv* **2019**, arXiv:1902.09314.
42. Zhang, C.; Li, Q.; Song, D. Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv* **2019**, arXiv:1909.03477.
43. Fei, H.; Li, B.; Liu, Q.; Bing, L.; Li, F.; Chua, T.S. Reasoning implicit sentiment with chain-of-thought prompting. *arXiv* **2023**, arXiv:2305.11255.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Electronics Editorial Office
E-mail: electronics@mdpi.com
www.mdpi.com/journal/electronics

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-6419-5