LLM-Augmented Contrastive Learning for Misinformation Detection in Social Networks

Hang Shen, *Member, IEEE*, Xiang Li, Xu Wang, Yuanfei Dai, Tianjing Wang, *Member, IEEE*, and Guangwei Bai

***Abstract*—Misinformation detection in social networks faces challenges due to complex semantics, scarcity of labeled data, and rapidly evolving false narratives. To address these issues, we present large language model (LLM)-augmented contrastive learning (LACL), a novel framework that integrates LLMs with contrastive learning (CL) for robust and accurate misinformation detection. We begin with an LLM-driven social media data augmentation strategy, utilizing prompt orchestration to generate diverse yet semantically consistent misinformation samples. These augmented samples are integrated into a CL-based detector, where the semantic richness and diversity introduced by the LLM enhance the CL's discriminative feature extraction and predictive capability, thus improving generalization beyond the original training data. To align with CL's discriminative goal, we develop a contrastive loss-aware joint training and fine-tuning approach where CL's discriminative feature learning actively constrains the LLM's hallucinations and guides the quality of augmentation. Through this closed-loop optimization, the CL-based detector progressively absorbs latent semantic knowledge from the LLM, effectively overcoming semantic complexity and reducing erroneous generations. Experimental results on four benchmark datasets (Twitter15, Twitter16, Weibo, and PHEME) demonstrate that LACL outperforms mainstream deep learning methods and surpasses approaches that apply commercial LLMs for detection without task-specific adaptation. These results hold consistently across different backbone LLMs (qwen and llama), highlighting LACL's enhanced robustness, adaptability to varying language contexts, and superior generalization capability.**

***Index Terms*—Contrastive learning (CL), fine-tuning, large language models (LLMs), misinformation detection, social networks.**

I. Introduction

THE rapid growth of the Internet has made information dissemination more convenient and facilitated the spread of misinformation and rumors, leading to significant societal challenges, including public confusion and even panic [1], [2]. To curb the spread of rumors, social media platforms have responded with manual content reviews and account restrictions, but these measures are costly and prone to bias. To improve detection efficiency, deep neural networks (DNNs) have been widely adopted, utilizing supervised [3], [4] and semisupervised methods [5], [6].

Although supervised learning has achieved promising results, its heavy reliance on large-scale labeled datasets and its inability to adapt to rapidly evolving rumor patterns limit its long-term effectiveness. To address these issues, contrastive learning (CL), a promising self-supervised learning approach, has shown great potential in tasks such as semantic segmentation [7], text classification [8], and named entity recognition [9]. When combined with graph neural networks (GNNs) [10], [11], CL enhances misinformation detection by extracting features from complex data structures, improving the analysis of content propagation in social networks.

The emotions and opinions expressed by social media users are essential for assessing content authenticity [12]. User-generated content, such as microblog interactions, reveals how misinformation spreads and helps understand its dynamics within networks. However, existing CL-based methods, such as DropEdge [13], AdaEdge [14], and NodeAug [15], which integrate GNNs, focus on misinformation propagation and network structure, often overlooking emotional context and diverse opinions. Similarly, natural language processing (NLP) techniques such as synonym replacement [16] improve text-level representation but struggle to capture subtle emotions, sarcasm, and cultural cues, limiting their effectiveness in capturing emotional responses, underlying motivations, and subtle cues such as sarcasm or cultural context.

Large language models (LLMs) have transformed NLP by leveraging vast parameter scales and extensive pretraining to achieve superior language understanding and capture subtle semantic nuances [17], [18]. These capabilities enable LLMs to identify misinformation patterns, implicit logic, and inconsistencies, making them powerful tools for rumor detection. However, directly applying LLMs is prone to generating hallucinations [19], [20], producing plausible content but factually incorrect. This inherent risk of LLM hallucination is a critical challenge, especially when employing LLMs for data augmentation in sensitive domains such as misinformation detection, as it can lead to the generation of misleading or

Received 1 March 2025; revised 9 July 2025; accepted 11 August 2025. Date of publication 18 September 2025; date of current version 6 February 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 61502230, Grant 61501224, and Grant 62202221, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20201357 and Grant BK20220331, and in part by the Six Talent Peaks Project in Jiangsu Province under Grant RJFW-020. *(Corresponding author: Yuanfei Dai.)*

The authors are with the College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University, Nanjing 211816, China (e-mail: hshen@njtech.edu.cn; daiyuanfei@njtech.edu.cn; wangtianjing@njtech.edu.cn; bai@njtech.edu.cn).

Digital Object Identifier 10.1109/TCSS.2025.3599080

counterproductive training samples. LLMs are also limited by training data biases and struggle to adapt to rapidly evolving rumor content. To address these limitations, including the propensity for hallucination, LLMs' advanced feature extraction and semantic understanding capabilities are synergistically combined with CL. While CL serves as the core detector by learning robust representations, LLMs enhance performance through diverse, semantically aligned augmentation. Importantly, a CL-guided fine-tuning strategy mitigates hallucinations and aligns LLM outputs with CL objectives.

A. Challenges and Related Works

Despite the considerable potential, the collaboration between LLMs and CL faces several pressing challenges.

1) Leveraging LLMs to Overcome Data: Limitations in the CL framework existing detection datasets, often drawn from a single social platform, limit model generalization across multiple sources. Traditional CL data augmentation, such as synonym replacement and sentence rearrangement [16], can expand data volume but struggle to capture nuanced semantic features such as sarcasm, metaphors, and implicit suggestions. They may even generate misleading samples. Furthermore, rumor propagation is deeply influenced by social contexts and event backgrounds, which traditional augmentation techniques fail to simulate, limiting their ability to replicate subtle variations in collective behavior. Feng et al. [21] used bidirectional multilevel graph CL and data augmentation strategies to improve rumor detection. The approach faces challenges in capturing fine-grained semantic features within complex contexts. Its performance across diverse cultural backgrounds is limited due to the inability to preserve unique rumor styles and propagation characteristics. In [22], LLM-enhanced news reframing was used to inject stylistic diversity. Each news article was transformed into multiple stylistic variations during training to increase data diversity and help the student network learn more robust features, which can be seen as a form of unidirectional knowledge transfer from the LLM. However, efficient data utilization and effective collaboration between the teacher LLM and the student remain to be explored.

2) Deep Extraction of Representations With Limited Samples: Effectively extracting deep semantic representations in social media rumor detection remains challenging due to the scarcity of samples in current public datasets. Traditional data augmentation methods [13], [14], [15], [16] partially mitigate this issue but primarily focus on shallow text transformations, failing to achieve substantial semantic enhancement. Thus, CL frameworks may struggle to capture deep semantic features in rumor texts, such as discourse logic, argument structure, and emotional inclination, when constructing positive and negative sample pairs. Moreover, rumor texts often involve complex contextual dependencies and implicit semantic links, which require deep semantic understanding and representation learning. Liu et al. [23] proposed a fake news detection framework that enhances news graph representation by integrating content, emotional information, and dissemination structure using GNNs and edge-aware techniques. However, under low-sample conditions, the framework struggles to extract deep semantic and emotional features effectively, limiting its generalization and early detection accuracy. Thus, a key challenge is to extract deep semantic features from limited samples using advanced augmentation and optimized CL training.

3) Synergy of LLMs and CL: DNN models, including CL, can be integrated through joint training or model concatenation [24]. One straightforward approach is to leverage LLMs' language understanding capabilities for detection tasks. However, due to misalignment, these methods struggle to harness LLMs' language understanding strength and CL's feature extraction potential. Hu et al. [25] proposed a fake news detection method that utilizes ChatGPT as an advisor rather than a detector, providing multiperspective reasoning and guidance. CALRec [26], a sequential recommendation framework, uses two-stage LLM fine-tuning to align user interaction sequences and target items, enhancing model performance by maximizing positive sample similarity and minimizing negative sample similarity. Dong et al. [27] proposed an unsupervised LLM alignment method for information retrieval, utilizing proximal policy optimization (PPO) to optimize LLM parameters. Jiang et al. [28] introduced CL to enhance multimodal LLMs by treating hallucinated text as hard negatives. This brings nonhallucinated text and visual samples closer while separating nonhallucinated and hallucinated text. In [29], LLMs were used to extract keywords and assess their relationship weights through graph Laplacian learning to automatically construct a knowledge graph (KG). MiLk-FD [30] effectively integrates the semantic and structural features of news content with factual knowledge from multiple KGs, resulting in superior performance in fake news detection. While these approaches utilize LLMs as data augmentation and KG extraction tools, they lack direct interlinking and feedback mechanisms. Thus, a challenge remains in designing a sustainable optimization method that enables the synergy of LLMs and CL.

B. Contributions and Organization

To address the aforementioned challenges, we propose LLM-augmented contrastive learning (LACL), a framework designed to enhance the stability and accuracy of fake news detection in social networks. In this framework, the LLM's advanced language understanding and generation capabilities compensate for CL's feature extraction limitations, while CL's discriminative learning improves LLM fine-tuning and data augmentation. The key contributions are threefold.

1) An LLM-based data augmentation method is developed to overcome social network data limitations. This method leverages the LLM's knowledge to generate diverse and semantically consistent misinformation samples, expanding CL's training dimension.
2) A LLM-assisted feature extraction and label prediction method is proposed, leveraging the LLM's semantic understanding and sample generation capabilities to enhance CL's ability to capture deeper representations, thereby mitigating the challenges posed by data limitations.

3) To align the LLM with the CL's discriminative objective, we develop a sustainable joint training and fine-tuning strategy. The contrastive loss guides LLM fine-tuning and augmentation strategy updates, where the generated high-quality data is leveraged to strengthen CL's feature extraction and classification capabilities. This closed-loop optimization enables the CL-based detector to progressively extract implicit knowledge from the LLM, overcoming its limitations in handling complex semantics.

Experimental results on multiple mainstream social media datasets highlight LACL's effectiveness, robustness, and applicability. The research questions addressed include.

1) Can LACL consistently outperform the performance upper bound of CL baseline methods?
2) What impact do data augmentation rounds and base LLM parameter size have on detection performance?
3) Can multiround LLM fine-tuning continue to improve performance, and do marginal effects exist?

The remainder of this article is organized as follows. Section II introduces the LACL architecture, which includes data preprocessing and prompt engineering, feature extraction and label prediction, and CL-LLM alignment. Section III presents datasets, base LLM, and benchmark method selection, as well as experimental setup. Section IV analyzes the experimental results and highlights key findings. Finally, Section V concludes the article and discusses potential future directions.

II. Proposed Method

Fig. 1 illustrates the LACL architecture, a framework designed to detect false content in social networks by fostering a symbiotic relationship between a CL-based detection network and an LLM optimizer. This architecture, comprising a data preprocessing module, a prompt orchestrator, a CL-based detector, and an LLM optimizer, is founded on the principle that CL and the LLM can mutually enhance each other. The LLM's role in generating high-quality augmented data is crucial for overcoming data scarcity, while the CL network's discriminative capabilities are leveraged not only for misinformation detection but also to provide a strong feedback signal for refining the LLM's generation strategy, thereby minimizing issues such as LLM hallucinations. This iterative refinement process aims to improve the quality of LLM-generated samples, which, in turn, enhances the CL-based detector's unsupervised classification performance. Conversely, the improved classification performance, as reflected through the contrastive loss function, provides a clearer guiding signal for the LLM's augmentation strategy. The workflow for this collaboration is divided into the following three stages.

1) *Data preprocessing and prompt orchestration*: Raw data is scraped from social networks using web crawlers and formatted accordingly. The processed data is fed into the prompt orchestrator, which generates high-quality augmented data to expand the CL's training dimension.
2) *LLM-assisted feature extraction and label prediction*: Augmented data is paired with the original data and input into the CL network to extract features and predict labels.

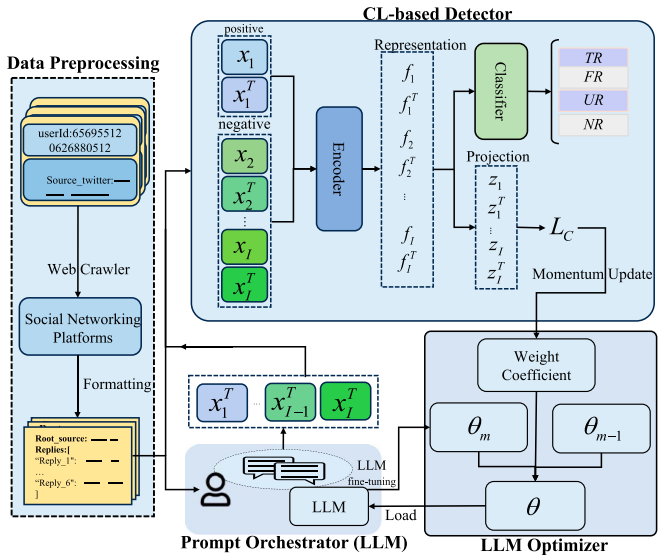


Fig. 1. LACL architecture.

After training, the CL model's parameters are frozen, and during testing, the classifier processes the extracted features to predict labels.

3) *Contrastive loss-aware joint fine-tuning and training*: The contrastive loss is leveraged to guide LLM fine-tuning, ensuring its outputs align with CL's discriminative capability. This alignment specifically improves the LLM's generation strategy and indirectly optimizes the feature extraction of the CL-based detection network.

A. Data Preprocessing and Prompt Orchestration

Fig. 2 illustrates the structure of each data instance, which typically includes the original post and its associated responses. The predefined prompt guides and constrains the LLM to ensure it can recognize and analyze typical characteristics of misinformation in social networks, such as logical incoherence, factual inaccuracies, strong emotional tone, and noticeable biases. Through this guidance, the LLM can capture subtle clues in the original data that may indicate content manipulation, thereby accurately identifying potential misinformation. While this prompt orchestration provides initial directional guidance to the LLM, the primary mechanism for robustly controlling semantic integrity and minimizing potential hallucinations during the augmentation process is the subsequent CL-guided LLM fine-tuning, as detailed in Section II-C. Once the analysis is complete, the LLM reorganizes and diversifies the original content to generate semantically consistent, varied augmented data.

The prompt orchestration should satisfy the following objectives.

1) *Holistic augmentation*: Augment data from a global perspective to effectively capture the dataset's overall features and distribution.
2) *Format consistency*: Ensure generated outputs adhere to the original data format, maintaining structural integrity for seamless integration.

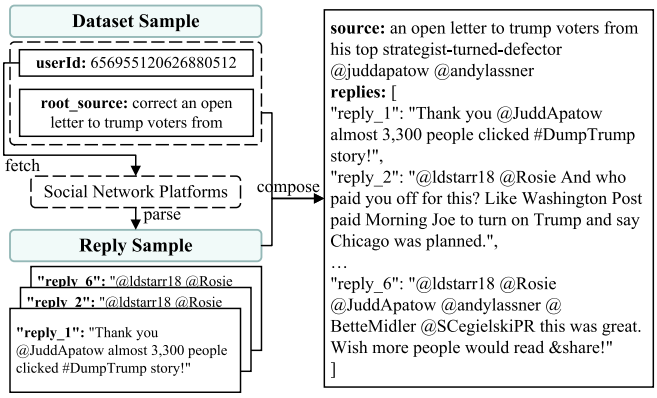


Fig. 2. Data preprocessing examples.

You are given data from a rumor detection dataset, and your task is to enhance the data based on the following instructions:
1. **Holistic Enhancement**:
- Enhance the data by considering its overall content, not just focusing on isolated parts. Make sure the enhancements reflect a comprehensive understanding of the entire data piece.
2. **Maintain Format Consistency**:
- Ensure the enhanced data retains the exact same format as the original data. No structural changes should occur.
3. **Increase Diversity**:
- Focus on making the data more diverse in terms of language expression, without altering the underlying meaning. The enhanced data should be more varied but semantically identical to the original.
4. **Preserve Semantics**:
- It is critical that the enhanced data preserves the exact same meaning as the original, even though the expressions may differ.

Fig. 3. Prompt orchestration example.

3) *Content diversity*: Encourage varied content generation, enhancing richness while preserving authenticity.
4) *Semantic alignment*: Maintain semantic consistency between the augmented and original data, preserving meaning critical for accurate misinformation detection.

Following these principles, we designed the prompt illustrated in Fig. 3, ensuring that each requirement is met in practice. As shown in Fig. 4, the augmented content not only restructures sentences as a whole to better capture global semantics. For example, rewriting "an open letter to Trump voters from his top strategist-turned-defector" as "A public letter to Trump supporters from his former strategist-turned-critic", but also preserves the original data format, including identical fields and hierarchical reply structures, ensuring compatibility with downstream processing. The language style is diversified while retaining the original meaning, such as changing "They love him" to "Theyre fully committed to him" for greater specificity. Moreover, the semantic core remains intact; for instance, "She obviously didnt look at all the havoc Drumpf has caused. He destroyed the USFL!" becomes "She clearly ignored all the damage Trump has already done. He ruined the USFL!", maintaining the critical tone and factual references while refining the phrasing for clarity. These examples collectively demonstrate that the prompt design effectively enhances data quality and diversity while ensuring structural and semantic consistency with the original dataset

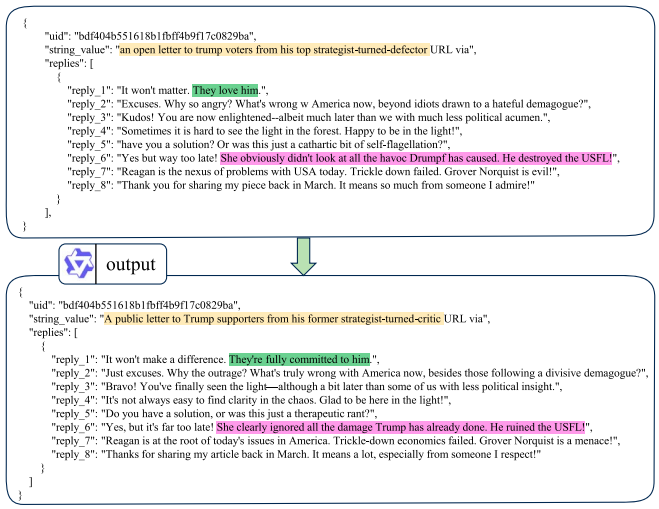


Fig. 4. Data augmentation example.

1) *Holistic augmentation*: This involves restructuring the sentence as a whole instead of relying on simple word replacements. For instance, "an open letter to Trump voters from his top strategist-turned-defector" is rewritten as "A public letter to Trump supporters from his former strategist-turned-critic," enhancing both the sentence structure and the richness of expression.
2) *Format consistency*: Both the original and augmented data retain identical fields (e.g., uid, string_value, and replies), with the hierarchical structure of replies preserved, ensuring compatibility for further processing and analysis.
3) *Diversity enhancement*: Language diversity is introduced while preserving the original meaning. For example, changing "They love him" to "They're fully committed to him" increases the specificity and richness of the expression.
4) *Semantic consistency*: The core meaning is preserved despite changes in expression. For example, "She obviously didn't look at all the havoc Drumpf has caused. He destroyed the USFL!" is rewritten as "She clearly ignored all the damage Trump has already done. He ruined the USFL!" This retains the critical tone and specific facts while refining the phrasing for clarity.

This example demonstrates that the prompt design enhances data quality and diversity while maintaining structural and semantic consistency with the original data.

B. LLM-Assisted Feature Extraction and Label Prediction

A social network dataset containing I instances is represented as $\{x_1, x_2, \ldots, x_I\}$, where x_i denotes the ith instance, with each instance processed by the LLM. The transpose of the feature vector for the ith instance is represented as x_i^T, which is used in various operations such as similarity computation and feature projection. Using a predefined prompt, the LLM generates augmented data that maintains semantic consistency but differs in expression. The CL detector comprises a feature extraction network based on bidirectional

encoder representations from transformers (BERT) [31] and a projection head built on a multilayer perceptron [32] for feature extraction. The feature representation of x_i is denoted as f_i. Let W_1 represent the weight matrix of the first layer of the MLP, which maps input features f_i to the hidden layer, and W_2 represent the weight matrix of the second layer, which maps the hidden layer output to the final representation z_i. This process is expressed as a nonlinear transformation, $z_i = g(f_i)$. Let $\sigma(\cdot)$ denote the activation function. The transformation can be written as

$$z_i = W_2\sigma(W_1 f_i). \tag{1}$$

In the projection space, the similarity function to measure the similarity between z_i and $z_{i'}$ is defined as

$$\text{sim}(z_i, z_{i'}) \triangleq \frac{z_i \cdot z_{i'}}{\|z_i\| \, \|z_{i'}\|}. \tag{2}$$

The samples involved in the loss calculation include both the original samples and their augmented counterparts, resulting in a total of $2I$ samples. The projection representation of x_i^T is denoted as z_i^T. Applying (2) to the InfoNCE Loss [33], the contrastive loss at the nth epoch is described as

$$\mathcal{L}_n = -\frac{1}{2I}\sum_{i=1}^{I} \log \frac{\exp\left(\text{sim}(z_i, z_i^T)/\tau\right)}{\sum_{i'=1 i' \neq i}^{2I} \exp\left(\text{sim}(z_i, z_{i'})/\tau\right)} \tag{3}$$

where τ is the temperature hyperparameter. Minimizing (3) encourages samples from the same class to cluster closely in the projection space while increasing the distance between samples of different classes. By utilizing the contrastive loss as a potent feedback signal, the LLM is guided to generate augmented samples that are not only diverse but also discriminatively valuable for the CL detector. This process inherently penalizes and reduces the generation of factually inconsistent, semantically drifting, or otherwise misleading augmented data that could be characterized as hallucinations.

The classifier in the CL detector is defined as $f : G \to Y = \cup_{c\in\{1,2,\ldots,C\}} y_c$, where C represents the number of label categories and the labels $y_c \in \{\text{NR}, \text{FR}, \text{TR}, \text{UR}\}$ with nonrumor (NR), false rumor (FR), true rumor (TR), and unverified rumor (UR). This classifier comprises a fully connected layer followed by a softmax activation function. During the testing phase, the extracted feature representations are fed into the classifier for category prediction.

In each batch, we treat x_1 and its augmented counterpart x_1^T as positive samples, since they are semantically similar, both stemming from the same original instance but with varied expressions. $\{x_2, \ldots, x_I\}$ and their augmented versions $\{x_2^T, \ldots, x_I^T\}$ are considered negative samples as they originate from different instances and hence are semantically dissimilar. The CL network's feature extractor processes these positive and negative pairs to generate their initial feature representations. As described in (1), the features are mapped into a shared feature space to optimize data distribution, ensuring that semantically similar points are brought closer, while dissimilar ones are pushed apart. Cosine similarity is used to quantify the similarity between feature vectors.

Once the feature extraction training is completed, the feature extractor is frozen. During the testing phase, new data is input into this frozen feature extractor. The extracted features are passed into the pretrained classifier, which maps them to the predefined categories of rumors. The final prediction outputs the labels, completing the rumor classification process.

C. Contrastive Loss-Aware Joint Fine-Tuning and Training

In this article, the LLM is strategically and progressively employed to enhance and diversify social network datasets rather than being directly involved in feature extraction or classification. This section introduces a joint fine-tuning and training mechanism. The augmented data generated by the LLM continually expands and enriches the training dataset, strengthening the CL detector's feature extraction and classification in subsequent training rounds. This iterative process ensures the LLM aligns with the CL detector's objectives.

Mainstream LLM fine-tuning methods include full fine-tuning [34], prompt learning [35], and parameter-efficient fine-tuning (PEFT) [36]. Among PEFT methods, low-rank adaptation (LoRA) [37] achieves efficient parameter updates by adding low-rank decomposition matrices alongside the original weight matrices. Based on LoRA, we introduced a bootstrapped LLM fine-tuning method for data augmentation.

Let θ_0 denote the initial LLM. Before the mth fine-tuning, θ_{m-1} is used for data augmentation, implying that the dataset and model parameters evolve iteratively through the fine-tuning process. The entire training process consists of N epochs, and the LLM is fine-tuned periodically at the end of the nth epoch. Denote the dataset at the end of the nth epoch after the mth fine-tuning as $\mathcal{D}_{n,m}$. Based on $\mathcal{D}_{n,m}$ and θ_0, the LLM fine-tuned in the mth iteration is represented as

$$\theta_m = \text{LoRA}\left(\mathcal{D}_{n,m}, \theta_0\right). \tag{4}$$

The update vector is computed as $\vartheta_{m-1} = \theta_{m-1} - \theta_0$ (similarly, $\vartheta_m = \theta_m - \theta_0$). Each entry in ϑ_{m-1}, which exists in a $\dim(\theta_0)$ dimensional space, can be considered an axis, where the sign of the parameter indicates the direction along the axis. Consequently, ϑ_{m-1} can be decomposed into a sign vector $\gamma_{m-1} \in \mathbb{R}^{\dim(\theta_0)}$ and a magnitude vector $\eta_{m-1} \in \mathbb{R}^{\dim(\theta_0)}$, expressed as $\vartheta_{m-1} = \gamma_{m-1} \odot \eta_{m-1}$, where $\odot$ denotes element-wise multiplication. Formally, $\gamma_{m-1} = \text{sgn}(\vartheta_{m-1})$ and $\eta_{m-1} \triangleq |\vartheta_{m-1}|$. $\text{sgn}(\vartheta_{m-1})$ returns $+1$, 0, or -1 depending on the sign of ϑ_{m-1}, and $\text{sgn}(\vartheta_{m-1}) \cdot |\vartheta_{m-1}| = \vartheta_{m-1}$.

The joint fine-tuning process, detailed in Algorithm 1, consists of the following three key steps.

1) *Trimming*: ϑ_{m-1}, to be merged, is first trimmed to remove redundant values, yielding $\hat{\vartheta}_{m-1}$. To eliminate redundancy, the top q% of the magnitudes in ϑ_{m-1} are retained, while the rest are set to 0 (see line 3). Subsequently, $\hat{\vartheta}_{m-1}$ is decomposed into the sign vector $\hat{\gamma}_{m-1}$ and the magnitude vector $\hat{\eta}_{m-1}$ (see line 4). A similar procedure is applied to θ_m.
2) *Sign election*: For each eth entry in $\hat{\gamma}_{m-1}$, $\hat{\vartheta}_{m-1}$, and $\hat{\eta}_{m-1}$, the values are denoted as $\hat{\gamma}_{m-1}^{(e)}$, $\hat{\vartheta}_{m-1}^{(e)}$, and $\hat{\eta}_{m-1}^{(e)}$.

Algorithm 1: Merge (θ_m,θ_{m-1},θ_0, λ_{m-1}, λ_m, α).

Input: θ_m, θ_{m-1}, θ_0, λ_{m-1}, λ_m, α, q
Output: θ
1 **foreach** $r \in \{m-1, m\}$ **do**
2 $\quad \vartheta_r \leftarrow \theta_r - \theta_0$;
3 $\quad \hat{\vartheta}_r \leftarrow Top(\vartheta_r, q)$;
4 $\quad \hat{\vartheta}_r \leftarrow \hat{\gamma}_r \odot \hat{\eta}_r$;
5 $\quad \gamma_r \leftarrow \text{sgn}(\hat{\vartheta}_r)$;
6 $\quad \eta_r \leftarrow |\hat{\vartheta}_r|$;
7 $\quad$ **for** $e \in dim(\theta_0)$ **do**
8 $\quad\quad \gamma^{(e)} \leftarrow \text{sgn}(\hat{\vartheta}^{(e)}_{m-1} + \hat{\vartheta}^{(e)}_m)$;
9 $\quad\quad \mathcal{R}^{(e)} \leftarrow \{r \in \{m-1, m\} | \hat{\gamma}^{(e)}_r = \gamma^{(e)}\}$;
10 $\quad\quad \vartheta^{(e)} \leftarrow \frac{1}{|\mathcal{R}^{(e)}|} \sum_{r \in \mathcal{R}^{(e)}} \lambda_r \hat{\vartheta}^{(e)}_r$;
11 $\theta \leftarrow \theta_0 + \alpha\vartheta$;
12 **return** θ;

Let ϑ denote the aggregated task vector, with its sign vector as γ. Resolving sign conflicts between θ_{m-1} and θ_m is a prerequisite for their merging (line 8). To achieve this, the eth entry of γ is computed as

$$\gamma^{(e)} = \text{sgn}(\hat{\vartheta}^{(e)}_{m-1} + \hat{\vartheta}^{(e)}_m). \tag{5}$$

3) *Weighted disjoint merge*: For the eth parameter in ϑ, denoted as $\hat{\vartheta}^{(e)}_r$, only the value from the model with a sign consistent with $\gamma^{(e)}$ is retained. Denoted by $\mathcal{R}^{(e)} = \{r \in \{m-1, m\} | \hat{\gamma}^{(e)}_r = \gamma^{(e)}\}$ the index set. The eth parameter of ϑ is computed as

$$\vartheta^{(e)} = \frac{1}{|\mathcal{R}^{(e)}|} \sum_{r \in \mathcal{R}^{(e)}} \lambda_r \hat{\vartheta}^{(e)}_r \tag{6}$$

where $\theta^{(e)}_r$ belongs to $\{\theta^{(e)}_{m-1}, \theta^{(e)}_m\}$ and λ_r is θ_r's weight, determined by the contrastive loss. The merged LLM is expressed as

$$\theta = \theta_0 + \alpha\vartheta, \vartheta = [\hat{\vartheta}^{(1)}, \hat{\vartheta}^{(2)}, ..., \hat{\vartheta}^{(dim(\theta_0))}]^T \tag{7}$$

with α being a hyperparameter.

The alignment process of the LLM with CL is integrated into joint training, as summarized in Algorithm 2. In this setup, LLM-driven augmentation is applied every T epochs, resulting in a total of $W = \lceil (N/T) \rceil$ augmentations over N epochs. Throughout the training, the LLM's fine-tuning is guided by the contrastive loss. This guidance is pivotal in reducing LLM hallucination outputs, ranging from overt factual errors to subtle misleading nuances in the augmented data by systematically penalizing generations that degrade the CL network's discriminative performance. This iterative process continuously prompts the LLM to refine its augmentation strategies towards producing high-quality, semantically faithful, and beneficial samples.

This contrastive loss is also central to our hallucination mitigation strategy. If the LLM produces augmented data that is a product of hallucination (e.g., containing factual inaccuracies,

Algorithm 2: Alignment (θ_0, θ_m, θ_{m-1}, λ_{m-1}, λ_m, α, $\mathcal{L}_n$, $\mathcal{D}_{n,m}$, m, n).

Input: θ_0, θ_m, θ_{m-1}, λ_{m-1}, λ_m, α, $\mathcal{L}_n$, $\mathcal{D}_{n,m}$, M, N, m, n
Output: θ
1 **if** $n \leq N$ **then**
2 $\quad$ **foreach** $n \in \{1, 2, \ldots, N\}$ **do**
3 $\quad\quad \varphi_n \leftarrow \mu\mathcal{L}_n + (1-\mu)\frac{1}{n-1}\sum_{n'=1}^{n-1}\mathcal{L}_{n'}$;
4 $\quad\quad \lambda_m \leftarrow \frac{\varphi_m}{\sum_{m'=1}^{m}\varphi_{m'}}$;
5 $\quad \theta_m \leftarrow LoRA\,(\mathcal{D}_{n,m}, \theta_0)$;
6 $\quad \theta \leftarrow$ Merge $(\theta_{m-1}, \theta_m, \theta_0, \lambda_{m-1}, \lambda_m, \alpha, q)$;
7 $\quad \theta_{m-1} \leftarrow \theta$;
8 $\quad \mathcal{D}_{n,m} \leftarrow$ Train (θ);
9 $\quad$ **if** $m \leq M$ **then**
10 $\quad\quad$ Alignment (θ_0, θ_m, θ_{m-1}, λ_{m-1}, λ_m, α, $\mathcal{L}_n$, $\mathcal{D}_{n,m}$, m, n);
11 $\quad\quad \lambda_{m-1} \leftarrow \lambda_m$;
12 $\quad\quad m \leftarrow m+1$;
13 $\quad$ **else**
14 $\quad\quad$ **return**;

deviating semantically from the original sample's class, or introducing misleading characteristics that obscure true class features), such samples will likely be poorly discriminated by the CL network. This poor discrimination translates to a higher Ln. This error signal is then directly used to adjust the LLM's parameters during the fine-tuning phase (Algorithms 1 and 2), effectively steering the LLM away from generating such problematic augmentations in subsequent rounds. The loss value, $\mathcal{L}_n$, as in (3), quantifies the CL model's ability to learn distinguishing misinformation features and reflects the quality of the LLM-augmented data. A smaller $\mathcal{L}_n$ indicates that the augmented data is semantically rich and discriminative, allowing the CL model to effectively distinguish between positive and negative samples; otherwise, performance degrades. By monitoring $\mathcal{L}_n$, the algorithm dynamically adjusts the adapter parameter fusion during the predefined fine-tuning rounds. To prevent the undue impact from $\mathcal{L}_n$ in a single epoch, a momentum update strategy [38] is employed to evaluate the LLM's augmentation effect (see lines 3–4), expressed as

$$\varphi_n = \mu\mathcal{L}_n + (1-\mu)\frac{1}{n-1}\sum_{n'=1}^{n-1}\mathcal{L}_n \tag{8}$$

where μ is the influence factor. Based on φ_n, the weight for the mth fine-tuning is adjusted as

$$\lambda_m = \frac{\varphi_m}{\sum_{m'=1}^{m}\varphi_{m'}}. \tag{9}$$

By calling LoRA (line 5), a customized LLM, θ_m, is generated, and then, by calling Algorithm 1, the fine-tuned model is integrated to produce an optimized new LLM (line 6). M

determines the maximum LLM fine-tuning rounds (lines 10–15). When $m > M$, the algorithm halts fine-tuning. It can be manually adjusted as needed.

This alignment process establishes a robust, mutually reinforcing feedback loop between the LLM and the CL network. The LLM's advanced semantic understanding and generation capabilities are harnessed to produce diverse augmented data, crucial for enhancing the CL network's training and its ability to capture complex data semantics. Concurrently, the CL network's loss function provides a direct and quantitative measure of the augmented data's quality and utility, effectively guiding the LLM's fine-tuning process. This closed-loop optimization is instrumental in progressively reducing LLM hallucinations and improving the overall quality of LLM-generated data, as the LLM is iteratively steered to produce samples that bolster the CL's discriminative power. Consequently, the enhanced LLM augmentation strategy indirectly leads to improved unsupervised classification performance by the CL network. This symbiotic cycle ensures that both components evolve to overcome their respective limitations, driving continuous optimization of the entire detection framework.

III. Experimental Preparation

A. Dataset Selection

In the experiments, four benchmark datasets widely used in social media misinformation detection were selected: Twitter15 [39], Twitter16 [39], Weibo [40], and PHEME [41]. Specifically, Twitter15 and Twitter16 contain 1490 and 818 source tweets, respectively, and support four-class rumor detection, including TR, FR, NR, and UR categories. The Weibo and PHEME datasets support binary classification, consisting of 9128 and 5922 instances, respectively. The Weibo dataset includes 4640 samples labeled as false and 4488 as true, while PHEME includes 3006 False and 2916 true instances. To enhance contextual richness for inference, original posts along with all associated replies were collected via web crawling from the corresponding social media platforms. For data partitioning, a standard split was adopted: 70% for training, 20% for testing, and 10% for validation, ensuring a comprehensive and reliable evaluation setting.

B. Comparative Methods

To comprehensively evaluate the proposed framework, eight representative algorithms were selected as baselines.

1) BiGCN [42]: Employs GNNs to capture rumor diffusion in both top-down and bottom-up directions.
2) BiMGCL [21]: Leverages a bidirectional graph structure and multilevel CL to model rumor propagation, reducing dependence on labeled data and improving detection through diverse graph structures.
3) LSTM [43]: Captures long-range dependencies via gating mechanisms, effective for sequential and semantic rumor modeling.
4) Text-CNN [44]: Employs CNN with multiscale convolutional filters to extract discriminative local semantic patterns efficiently for text classification.
5) RCNN [45]: Combines CNN-based local feature extraction and RNN-based global context modeling to effectively detect rumors.
6) HAN [46]: Utilizes dual-level attention mechanisms (word and sentence) to capture hierarchical text features, enhancing rumor detection interpretability.
7) ELKP [47]: Enhances language models via knowledge-driven prompting and external knowledge injection, improving semantic reasoning and adaptability for misinformation detection.
8) SRD-PSID [48]: Leverages contrastive self-supervised learning of heterogeneous social and semantic patterns, enriching rumor representation and detection accuracy.

To assess the impact of different strategies on performance, we examine the framework across three key dimensions: the choice of base LLM (qwen [49] versus llama [50]), the number of data augmentation rounds (1–3), and the number of fine-tuning rounds (none, 1, or 2). These dimensions are critical for understanding how varying LLM sizes and augmentation strategies influence detection. Table I summarizes nine proposed configurations based on these factors.

1) Proposed-1, -2, and -3: Utilize qwen-7B as the base LLM, with 1, 2, and 3 data augmentation rounds, respectively, and no fine-tuning.
2) Proposed-4 and -5: Also use qwen-7B, but add 1 or 2 fine-tuning rounds in addition to data augmentation.
3) Proposed-6 and -9: Uses a larger qwen-14B base LLM with 3 data augmentation rounds to assess the impact of a larger model on performance.
4) Proposed-7 and -8: Employ llama models with size of 7B, exploring different combinations of data augmentation rounds and fine-tuning times.

C. Experimental Design and Configuration

We examined the impact of data augmentation, LLM fine-tuning, and the parameter size of the base LLM. The first experiment utilized CL baselines to evaluate LACL's performance boundary. The second focused on analyzing how augmentation rounds influence detection performance. The third assessed the impact of fine-tuning rounds (representing training costs). The 7B-parameter base LLM served as the primary subject in all experiments, while the 14-parameter base LLM, with three augmentation rounds, was introduced to explore the performance upper bound.

Two widely recognized evaluation metrics were employed to quantitatively analyze and assess the performance of rumor detection methods in social networks, i.e.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

and

$$F1 = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}. \tag{11}$$

TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. Accuracy provides an intuitive measure of correctly identified instances'

TABLE I
CATEGORIZATION OF PROPOSED METHODS

Classification of Proposed Methods	Choices of Base LLM and Model Size				Data Augmentation Rounds and Fine-Tuning Times (W, M)				
	Qwen-7B	Qwen-14B	Llama-7B	Llama-13B	$(1,0)$	$(2,0)$	$(3,0)$	$(3,1)$	$(3,2)$
Proposed-1	✓				✓				
Proposed-2	✓					✓			
Proposed-3	✓						✓		
Proposed-4	✓							✓	
Proposed-5	✓								✓
Proposed-6		✓					✓		
Proposed-7			✓		✓				
Proposed-8			✓					✓	
Proposed-9				✓			✓		

TABLE II
DEFAULT PARAMETER SETTINGS

Parameter	Value
Batch size	256
Training epochs (N)	25
Augmented cycles (T)	10
Magnitude filtering ratio (q)	Top 20%
LLM-fusion hyperparameter (α)	1
Influence factor (μ)	0.99
Maximum LLM fine-tuning rounds (M)	2
LoRA rank	8
LoRA learning rate	5×10^{-5}
LoRA training epochs	5

TABLE III
GPT-4O DIRECT DETECTION ACCURACY ON TWITTER15 AND TWITTER16 (BY RUMOR TYPE)

Dataset	TR(%)	NR(%)	FR(%)	UR(%)	Avg Acc(%)
Twitter15	37.15	31.89	42.36	39.04	37.61
Twitter16	32.18	38.94	41.57	43.09	38.94

TABLE IV
GPT-4O DIRECT DETECTION ACCURACY ON WEIBO AND PHEME (BINARY CLASSIFICATION)

Dataset	True(%)	False(%)	Avg Acc(%)
Weibo	58.37	45.92	52.15
PHEME	63.45	49.18	56.32

proportions, while the F1 score emphasizes the balance between precision and recall, highlighting the proportion of correctly classified instances.

Key hyperparameters, learning rate, training epochs, and batch size were kept consistent throughout the experiments to ensure a fair evaluation. Detailed training parameters were set as shown in Table II. This value of q controls the magnitude filtering process, where the top 20% of the most significant features are selected during training. A value of $\alpha = 1$ indicates a balanced model fusion. In light of the dataset size and label distribution, LoRA hyperparameters were selected to promote robust convergence and generalization. A learning rate of 5×10^{-5} was selected for LoRA (rank $= 8$) to balance gradient stability and task adaptation.

Tests ran on a high-performance server featuring an Intel Core i9-14900 K processor, 64 GB DDR5 5200 MHz RAM, a 4 TB PCIe 4.0 SSD, an ASUS PRIME Z790-P WIFI D5 motherboard, and dual Gigabyte RTX 4090 24 GB GPUs.

IV. RESULT ANALYSIS

A. Performance Bound Analysis

Note that the proposed LACL framework does not rely on directly using LLMs for misinformation detection. Instead, it leverages the semantic understanding and generation capabilities of LLMs to augment the CL-based detector in a task-aware, domain-adaptive manner. To better contextualize the performance bounds of our method, we first conducted a preliminary study by directly applying a mainstream LLM, GPT-4o,[1] to detect misinformation in social media content. Specifically, we randomly sampled 200 instances from each of four representative datasets, Twitter15, Twitter16, Weibo, and PHEME, and directly input them into GPT-4o for binary classification (i.e., true or false). As shown in Table III, the performance of GPT-4o on Twitter15 and Twitter16 is limited, with average accuracies of only 37.61% and 38.94%, respectively. The model exhibited considerable inconsistency across the four rumor types, suggesting difficulty in handling category-specific nuances in English social media content. Table IV shows results on the Weibo and PHEME datasets under binary classification. While performance is slightly improved compared to Twitter-based datasets, average accuracies of 52.15% (Weibo) and 56.32% (PHEME) still reflect suboptimal generalization, especially given the complexity of linguistic and cultural variations present in cross-lingual misinformation. These results highlight that even advanced LLMs lack the task-specific discriminative capacity for robust misinformation detection in cross-lingual and culturally nuanced contexts.

[1]https://chatgpt.com/

TABLE V
PERFORMANCE COMPARISON ON TWITTER15

Method	ACC(%)	F1				Avg F1(%)
		TR(%)	NR(%)	FR(%)	UR(%)	
BiGCN	74.52	80.07	68.10	76.45	71.04	73.92
BiMGCL	71.45	76.97	66.87	72.99	68.14	71.24
LSTM	76.43	85.20	66.10	75.77	75.34	75.60
Text-CNN	77.45	87.50	68.70	**79.48**	74.19	77.47
RCNN	73.40	80.70	64.46	72.94	72.72	72.71
HAN	76.79	86.22	68.33	76.89	75.70	76.79
ELKP	75.70	73.21	74.66	72.48	75.10	73.86
Proposed-1	78.11	**90.68**	69.42	75.73	74.12	77.48
Proposed-6	**81.81**	89.57	**80.29**	78.43	**78.01**	**81.57**
Proposed-7	78.03	89.52	70.65	76.49	73.18	77.46
Proposed-8	80.37	89.35	78.13	77.82	75.93	80.31
Proposed-9	82.34	88.97	75.78	83.06	80.84	82.16

TABLE VI
PERFORMANCE COMPARISON ON TWITTER16

Method	ACC(%)	F1				Avg F1(%)
		TR(%)	NR(%)	FR(%)	UR(%)	
BiGCN	76.42	86.09	66.68	73.23	75.82	75.45
BiMGCL	75.30	84.79	62.20	75.32	76.36	74.66
LSTM	75.92	90.14	71.11	64.36	81.57	76.80
Text-CNN	71.45	76.97	66.87	72.99	68.14	71.24
RCNN	76.54	90.41	71.26	71.26	77.50	77.06
HAN	77.16	87.17	70.88	70.88	**80.00**	77.20
ELKP	77.41	76.35	74.40	74.40	76.77	75.77
Proposed-1	79.63	94.59	77.64	71.26	76.92	80.10
Proposed-6	84.56	94.59	78.72	76.92	89.74	84.99
Proposed-7	79.91	93.87	78.92	72.13	76.45	80.34
Proposed-8	85.08	95.92	76.89	77.05	85.73	83.90
Proposed-9	85.39	95.45	80.26	79.87	87.89	85.87

Note: Bold values in tables V and VI are represent the numerical results, which indicate key performance inflection points or the upper and lower performance bounds.

Next, we evaluated the upper and lower performance bounds of the proposed method by comparing them with a range of representative baselines across the Twitter15 and Twitter16 datasets. As shown in Table V, on Twitter15, the proposed-1 variant achieved an accuracy of 78.11%, surpassing BiGCN (74.52%) and BiMGCL (71.45%) by 3.59% and 6.66%, respectively. Compared to traditional sequential and convolutional models such as LSTM (76.43%), RCNN (73.40%), and text-CNN (77.45%), proposed-1 also demonstrated consistent improvements of 1.68%, 4.71%, and 0.66%, respectively. When compared with the attention-based HAN (76.79%) and knowledge-enhanced ELKP (75.70%), proposed-1 yielded accuracy gains of 1.32% and 2.41%. The performance of the llama-based variants was even more notable. Proposed-6 achieved an accuracy of 81.81%, outperforming all baselines by a substantial margin: +7.29% over BiGCN, +10.36% over BiMGCL, +5.38% over LSTM, +4.36% over Text-CNN, +8.41% over RCNN, +5.02% over HAN, and +6.11% over ELKP. Proposed-7 and -8, which use llama-7B with different training strategies, further verified this trend with accuracies of 78.03% and 80.37%, respectively, while proposed-9 (based on llama-13B) achieved the highest performance at 82.34%, exceeding even proposed-6 by 0.53%. These results demonstrate the effectiveness of our LLM-augmented contrastive learning approach and the scalability of LACL across different base model sizes and configurations.

Similarly, as shown in Table VI, on Twitter16, proposed-1 reached an accuracy of 79.63%, outperforming BiGCN (76.42%) and BiMGCL (75.30%) by 3.21% and 4.33%, respectively. It also surpassed LSTM (75.92%), RCNN (76.54%), Text-CNN (71.45%), HAN (77.16%), and ELKP (77.41%) by 3.71%, 3.09%, 8.18%, 2.47%, and 2.22%, respectively. Among the llama-based methods, proposed-6 achieved an accuracy of 84.56%, leading all baseline models with clear margins: +8.14% over BiGCN, +9.26% over BiMGCL, +8.64% over LSTM, +8.02% over RCNN, +7.40% over HAN, +7.15% over ELKP, and +13.11% over text-CNN. In addition, proposed-7 and -8 showed promising performance, reaching 79.91% and 85.08%, respectively. Notably, proposed-9 obtained the best accuracy of 85.39%, with an Avg F1 of 85.87%, confirming the method's robustness and balanced classification performance across all rumor categories (TR, NR, FR, UR). These results highlight LACL's ability to scale effectively with stronger base LLMs and maintain superior performance across complex misinformation classification tasks.

For the FR category (rumors verified as false or inaccurate), in-depth analysis revealed a consistent performance gap between our proposed method and several baseline models, despite the overall superiority of the proposed variants. On Twitter15, proposed-1 achieved an F1 score of 75.73%, which was 0.72% lower than BiGCN and 3.75% lower than the best-performing Text-CNN (79.48%), and also fell behind models such as HAN and LSTM. Similarly, on Twitter16, BiGCN and BiMGCL surpassed proposed-1 by 1.97% and 4.06%.

This gap could be attributed to the dual-effect nature of LLM-based data augmentation. While LLMs improved generalization by generating diverse and fluent samples, they were not inherently equipped to verify factual accuracy. During augmentation, LLMs may unintentionally dilute or even rationalize key features indicative of falsehood in FR-class samples. For example, content that originally contained extreme claims, inconsistencies, or manipulative phrasing might have been softened or rephrased, making the misinformation appear more credible. This process likely blurred the semantic boundary between false and true content, thereby undermining the distinctiveness of the FR category.

In the binary classification setting of the Weibo dataset, the proposed method demonstrated clear advantages over all baseline models. As shown in Table VII, proposed-6 achieved a breakthrough performance with an accuracy of 91.63% and an average F1 score of 91.62%, both exceeding the 90% threshold. Compared to the best-performing baseline in accuracy (RCNN, 84.23%) and in average F1 (Text-CNN, 83.88%), proposed-6 attained substantial improvements of +7.40% and +7.74%, respectively. Even the lighter proposed-1 variant, which applied only a single round of LLM-based augmentation, still surpassed all baselines with an accuracy of 85.30% and an average F1 of 87.80%, demonstrating the effectiveness of our augmentation strategy even under low-resource configurations.

TABLE VII
PERFORMANCE COMPARISON ON WEIBO

Method	ACC(%)	F1		Avg F1(%)
		False(%)	True(%)	
LSTM	80.84	80.64	81.02	80.83
Text-CNN	83.87	83.94	83.81	83.88
RCNN	84.23	84.38	83.07	83.73
HAN	82.23	81.30	83.04	82.17
SRD-PSID	83.22	83.09	83.24	83.16
Proposed-1	85.30	87.19	88.41	87.80
Proposed-6	91.63	91.42	91.82	91.62
Proposed-7	86.92	86.58	87.26	86.92
Proposed-8	89.47	89.32	89.61	89.47
Proposed-9	90.78	90.65	90.91	90.78

TABLE VIII
PERFORMANCE COMPARISON ON PHEME

Method	ACC(%)	F1		Avg F1(%)
		False(%)	True(%)	
LSTM	86.89	87.34	86.41	86.88
Text-CNN	85.37	86.08	84.59	85.34
RCNN	87.23	87.48	86.97	87.23
HAN	88.16	88.63	87.65	88.14
ELKP	87.20	87.52	86.08	86.80
Proposed-1	89.01	89.14	88.86	89.00
Proposed-6	91.89	92.21	91.53	91.87
Proposed-7	89.23	88.97	89.48	89.23
Proposed-8	90.87	91.43	90.31	90.87
Proposed-9	91.55	91.98	91.12	91.55

On the PHEME dataset, as shown in Table VIII, the proposed models maintained a strong lead, although the margin of improvement was relatively reduced. Proposed-1 achieved 89.01% accuracy and 89.00% average F1, slightly outperforming the best baseline, HAN (88.14%). Proposed-6 continued to lead across all metrics with 91.89% accuracy, 92.21% F1 for False, and 91.53% F1 for true, but the relative performance gap was narrower than that observed on the Weibo dataset.

We also examined the impact of base LLM choice on the performance of the proposed methods. As shown in Tables V–VIII, when llama was used as the base model, the corresponding methods (proposed-7, -8, and -9) consistently demonstrated superior performance across all four datasets. Due to differences in language-specific strengths among base models, llama outperformed qwen on the english-based Twitter datasets, while qwen led on the Chinese-based Weibo dataset. In the fine-tuned setting (proposed-4 and -8), qwen consistently yielded slightly better improvements than llama, suggesting that qwen offered stronger instruction-following capabilities for the detection task. In the non-fine-tuned configurations (proposed-1 and -7), the two base models exhibited a trade-off in performance. These results indicated that the proposed method is adaptable to different base LLMs and performs robustly across various model sizes, confirming its versatility and scalability.

TABLE IX
RESULTS ON TWITTER15 WITH DIFFERENT AUGMENTATION ROUNDS

Method	ACC(%)	F1				Avg F1(%)
		TR(%)	NR(%)	FR(%)	UR(%)	
Proposed-1	78.11	**90.68**	69.42	75.73	74.12	77.48
Proposed-2	78.45	86.06	**75.55**	75.49	75.52	78.15
Proposed-3	**79.12**	87.89	73.91	79.24	74.28	78.83
Proposed-6	**81.81**	89.57	**80.29**	78.43	78.01	81.57

Note: Bold values in tables IX to XIV are represent the numerical results, which indicate key performance inflection points or the upper and lower performance bounds.

TABLE X
RESULTS ON TWITTER16 WITH DIFFERENT AUGMENTATION ROUNDS

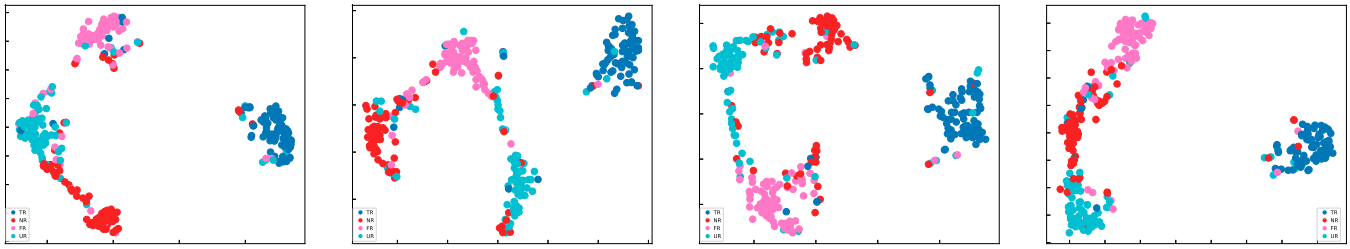
Method	ACC(%)	F1				Avg F1(%)
		TR(%)	NR(%)	FR(%)	UR(%)	
Proposed-1	79.63	**94.59**	77.64	71.26	76.92	80.10
Proposed-2	80.86	93.33	75	75.86	80.84	81.17
Proposed-3	**83.33**	94.28	76.59	78.16	87.67	84.18
Proposed-6	**84.56**	94.59	78.72	76.92	89.74	84.99

These findings suggested that while the proposed method generalized well across datasets, its performance was further influenced by the choice of base LLM and parameter scale. In high-context, semantically rich environments such as Weibo, qwen-based configurations demonstrated more substantial gains, particularly under fine-tuning settings. In contrast, llama-based models exhibited advantages on structurally constrained, english-language datasets such as PHEME. Moreover, larger LLMs generally yielded better detection accuracy, although they required increased computational costs. These results confirmed the LACL framework's adaptability to varying language settings, model capacities, and augmentation strategies, while consistently maintaining superior performance across diverse misinformation detection tasks.

B. Impact of Data Augmentation Rounds

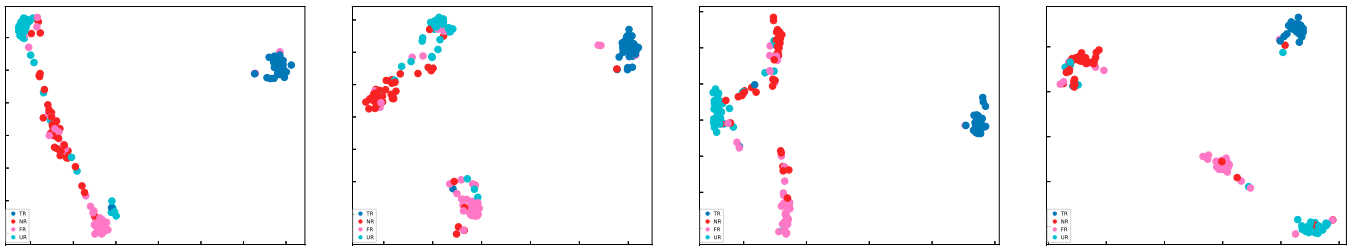
This experiment investigated the impact of data augmentation rounds on model detection performance. As shown in Table IX, the detection accuracy on Twitter15 improved gradually with additional augmentation rounds. Specifically, proposed-3 achieved a 0.67% and 1.01% accuracy increase compared to proposed-1 and -2, respectively. Similarly, as presented in Table X for Twitter16, proposed-3 improved accuracy by 2.47% and 3.7% compared to proposed-1 and -2, although it fell short of proposed-6, which achieved the highest accuracy of 84.56%. Feature visualizations further illustrated that increased augmentation rounds made class boundaries more distinct. This phenomenon could be attributed to two factors. First, as the number of augmentation rounds increased, the generated data samples became more diverse, allowing the model to learn more comprehensive features. On the other hand, larger parameter LLMs, with their superior language and contextual understanding, generated higher-quality samples, thereby enhancing detection performance.

However, category-specific detection results exhibited noticeable fluctuations. On Twitter15, proposed-1 achieved the best performance for the true rumor (TR) category. As shown in



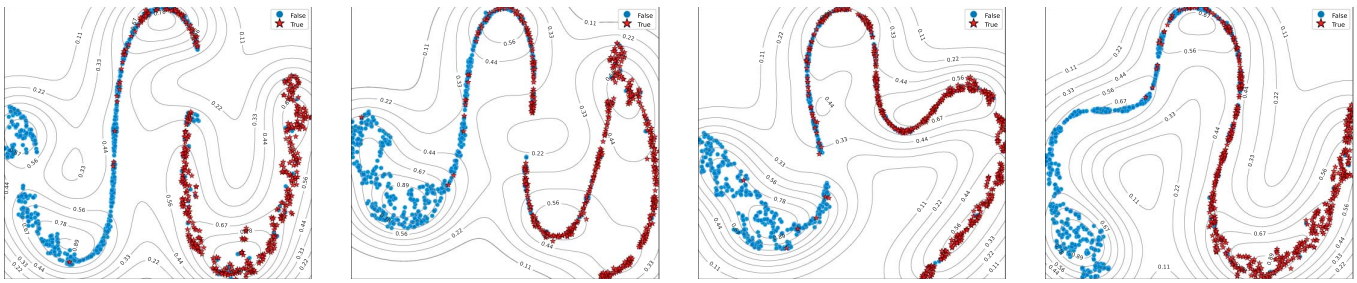
(a) (b) (c) (d)

Fig. 5. Feature distribution of different augmentation rounds on Twitter15. (a) Proposed-1. (b) Proposed-2. (c) Proposed-3. (d) Proposed-6.



(a) (b) (c) (d)

Fig. 6. Feature distribution of different augmentation rounds on Twitter16. (a) Proposed-1. (b) Proposed-2. (c) Proposed-3. (d) Proposed-6.



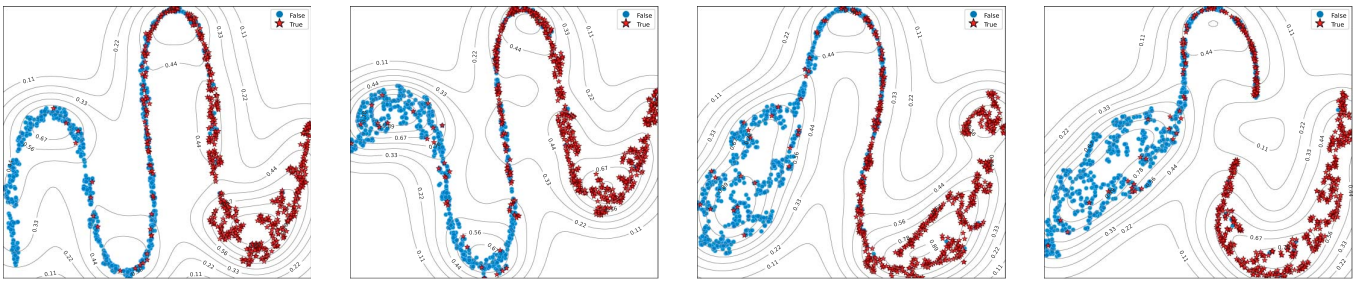
(a) (b) (c) (d)

Fig. 7. Feature distribution of different augmentation rounds on Weibo. (a) Proposed-1. (b) Proposed-2. (c) Proposed-3. (d) Proposed-6.

Fig. 5(a) and (b), the TR feature points of proposed-1 were more tightly clustered than those of proposed-2, indicating more stable class separation. For the nonrumor (NR) category, proposed-2 outperformed proposed-3, as illustrated in Fig. 5(b) and (c), yet both remained inferior to proposed-6, which exhibited better interclass separation. Notably, Fig. 5(c) revealed significant overlap between NR and unverified rumor (UR) feature points in proposed-3, reflecting ambiguity in feature space. Similar trends were observed on Twitter16 in Fig. 6(a)–(d), where proposed-6 consistently showed stronger feature discrimination across categories.

These fluctuations could be attributed to two main factors. First, the quality of LLM-augmented data varied across categories, making certain classes, such as NR and UR, more susceptible to misleading or noisy samples. Second, the augmentation process may have introduced features that conflicted with the original class semantics, reducing intraclass cohesion or causing overlap with adjacent classes. While LLM augmentation enhanced overall performance, its impact on fine-grained category boundaries remained sensitive to data quality and semantic alignment.

As shown in Table XI, increasing the LLM augmentation rounds from 1 to 3 improved classification accuracy on Weibo from 87.88% to 89.70%, narrowing the F1-score gap between True and False classes from 0.91% to 0.36%. Feature visualizations (Fig. 7) illustrated that False-class samples transitioned from a bimodal (proposed-1) to an unimodal distribution (proposed-2), with decreased overlap between classes. Although proposed-3 improved clustering compactness, the True class consistently retained a bimodal distribution. On PHEME (Table XII), accuracy improved from 89.01% to 90.61% across augmentation rounds. True and False class F1-scores increased to 90.20% and 90.99%, respectively. Feature distributions (Fig. 8) revealed similar clustering enhancements, shifting the False class from bimodal to unimodal distribution, and reducing interclass overlap. However, the True class again maintained its bimodal feature distribution.



(a) (b) (c) (d)

Fig. 8. Feature distribution of different augmentation rounds on PHEME. (a) Proposed-1. (b) Proposed-2. (c) Proposed-3. (d) Proposed-6.

TABLE XI
RESULTS ON WEIBO WITH DIFFERENT AUGMENTATION ROUNDS

Method	ACC(%)	F1		Avg F1(%)
		False(%)	True(%)	
Proposed-1	87.88	87.40	88.31	87.86
Proposed-2	88.62	88.30	88.93	88.62
Proposed-3	89.70	89.51	89.87	89.69
Proposed-6	**91.63**	91.42	91.82	91.62

TABLE XII
RESULTS ON PHEME WITH DIFFERENT AUGMENTATION ROUNDS

Method	ACC(%)	F1		Avg F1(%)
		False(%)	True(%)	
Proposed-1	89.01	89.14	88.86	89.00
Proposed-2	89.94	90.21	89.66	89.94
Proposed-3	90.61	90.99	90.20	90.60
Proposed-6	**91.89**	92.21	91.53	91.87

TABLE XIII
RESULTS ON TWITTER15 WITH DIFFERENT FINE-TUNING TIMES

Method	ACC(%)	F1				Avg F1(%)
		TR(%)	NR(%)	FR(%)	UR(%)	
Proposed-3	79.12	87.89	73.91	79.24	74.28	78.83
Proposed-4	80.13	88.88	**79.10**	76.64	75.71	80.08
Proposed-5	80.81	89.87	**71.83**	**84.21**	77.63	80.89
Proposed-6	81.81	89.57	80.29	78.43	78.01	81.58

TABLE XIV
RESULTS ON TWITTER16 WITH DIFFERENT FINE-TUNING TIMES

Method	ACC(%)	F1				Avg F1(%)
		TR(%)	NR(%)	FR(%)	UR(%)	
Proposed-3	83.33	94.28	76.59	**78.16**	**87.67**	84.18
Proposed-4	85.24	95.89	77.53	76.28	86.11	83.95
Proposed-5	85.31	95.91	78.29	**77.16**	**86.84**	85.55
Proposed-6	**84.56**	94.59	78.72	76.92	89.74	84.99

Overall, two conclusions could be drawn: 1) increased augmentation rounds enhanced sample diversity, aiding comprehensive feature learning; and 2) larger LLMs produced higher-quality augmented samples, thereby improving model performance, though larger parameter scales implied higher training and fine-tuning costs.

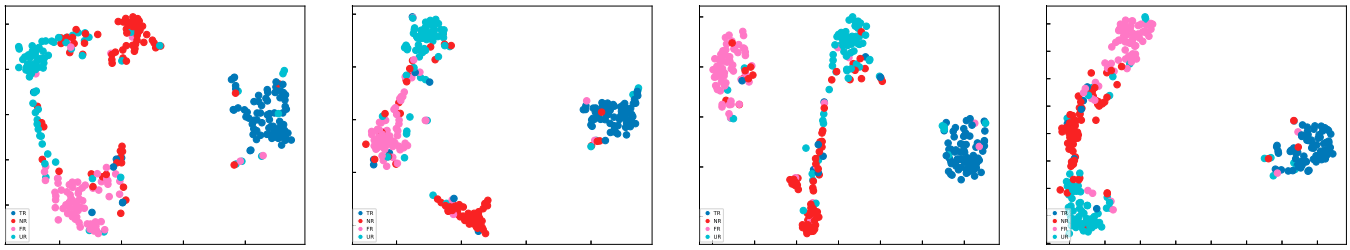
C. Impact of LLM Fine-Tuning Rounds

The experimental results above confirm that the proposed framework is highly adaptable to different base LLMs. In this section, we take qwen-based configurations (proposed-3, -4, -5, and -6) as the research objects and focus on investigating how the number of fine-tuning rounds influences detection performance.

As the number of fine-tuning rounds for qwen-7B increased, LLM-assisted CL detection accuracy improved, though it did not surpass the performance of proposed-6. Specifically, on Twitter15, as shown in Table XIII, F1 scores for the NR and FR categories fluctuated significantly. Compared to proposed-3, proposed-4 saw a 7.27% decrease in the NR F1 score but a 7.57% increase in the FR F1 score. Other categories generally showed an upward trend in F1 scores. Fig. 9 clarified these results: initially, the NR and FR categories overlapped significantly, which lowered the accuracy for FR. After two fine-tuning rounds, FR showed stronger clustering and more compact feature distribution, while *NR* experienced weaker clustering and increased overlap with the UR category.

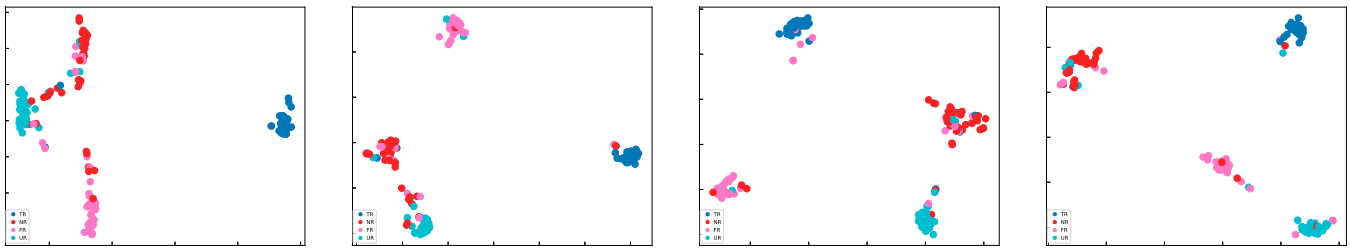
On Twitter16, as shown in Table XIV, the F1 scores for TR and NR categories steadily improved. For the FR category, proposed-4 and -5 achieved F1 scores 1.88% and 1% lower than proposed-3, respectively. Similarly, in the UR category, proposed-4 and -5 exhibited decreases of 1.56% and 0.83% compared to proposed-3. Fig. 10 illustrated that the clustering degree of feature points increased with the number of fine-tuning rounds; however, overlaps for FR, UR, and NR categories negatively impacted detection precision. This underscored the model's progressive optimization in feature extraction. The recognition of natural samples improved with iterations, yielding cumulative F1 score gains of 1.63% for TR and 1.7% for NR. Additionally, classification ambiguity persisted for boundary samples, particularly in overlapping regions of FR and UR categories, leading to slight performance degradation. F1 scores remained within a narrow range of 84%–85%, while UR metrics displayed an initial decline followed by a recovery.

As shown in Table XV, increasing the number of LLM fine-tuning rounds (from proposed-3 to -4 to -5) led to consistent performance improvements on the Weibo dataset. Specifically,



(a) (b) (c) (d)

Fig. 9. Feature distribution of different fine-tuning rounds on Twitter15. (a) Proposed-3. (b) Proposed-4. (c) Proposed-5. (d) Proposed-6.



(a) (b) (c) (d)

Fig. 10. Feature distribution of different fine-tuning rounds on Twitter16. (a) Proposed-3. (b) Proposed-4. (c) Proposed-5. (d) Proposed-6.

TABLE XV
RESULTS ON WEIBO WITH DIFFERENT FINE-TUNING TIMES

Method	ACC(%)	F1		Avg F1(%)
		False(%)	True(%)	
Proposed-3	89.70	89.51	89.51	89.69
Proposed-4	90.23	90.05	90.41	90.23
Proposed-5	90.45	90.14	90.41	90.33
Proposed-6	91.63	91.42	91.82	91.62

TABLE XVI
RESULTS ON PHEME WITH DIFFERENT FINE-TUNING TIMES

Method	ACC(%)	F1		Avg F1(%)
		False(%)	True(%)	
Proposed-3	90.61	90.99	90.20	90.59
Proposed-4	91.04	91.26	90.79	91.02
Proposed-5	91.29	91.61	90.96	91.28
Proposed-6	91.89	92.21	91.53	91.87

accuracy rose from 89.70% to 90.45%, while the F1-scores for the True and False classes increased to 90.41% and 90.14%, respectively, indicating enhanced class balance. Feature distribution visualizations in Fig. 11(a) revealed overlapping density contours (e.g., 0.56) between classes and a bimodal distribution in the True class. With one round of fine-tuning [Fig. 11(b)], this bimodality persisted in the core region (0.78–0.89). However, after two rounds [Fig. 11(c)], feature points became more concentrated in the lower-right quadrant, suggesting that fine-tuning refined LLM parameters and promoted tighter intraclass clustering in the feature space.

Similarly, on the PHEME dataset (Table XVI), fine-tuning (proposed-4 and -5) resulted in steady performance gains: accuracy improved from 90.61% to 91.29%, and the True/False class F1-scores increased from 90.20%/90.99% to 90.96%/91.61%. Visualization in Fig. 12 showed that transitioning from no fine-tuning (proposed-3) to one round (proposed-4) reduced class overlap in the central region. Further, Fig. 12(b) and (c) demonstrated that proposed-5 yielded more compact True-class clusters and sparser density contours in overlapping regions.

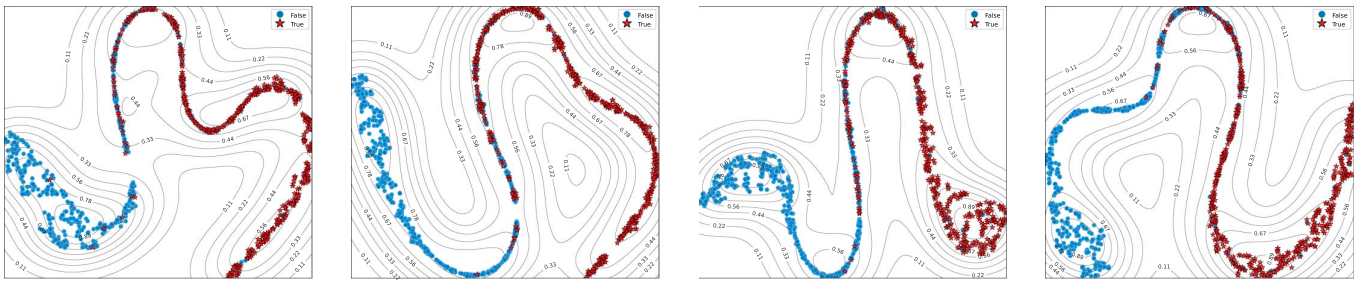
TABLE XVII
MEMORY USAGE AND INFERENCE MODE IN LLM DATA AUGMENTATION

Proposed Methods	Base LLM	Graphics Card Configuration	Average Memory Usage	Inference Mode
–1 to –5	Qwen-7B	Dual 4090	14.4GB per card	Data Parallelism
–6	Qwen-14B	Dual 4090	22.6GB per card	Model Parallelism
–7 and –8	Llama-7B	Dual 4090	14.6GB per card	Data Parallelism
–9	Llama-13B	Dual 4090	21.1GB per card	Model Parallelism

Notably, despite these improvements, the 14B-based proposed-6 model still outperformed both proposed-4 and -5 across all metrics on both datasets.

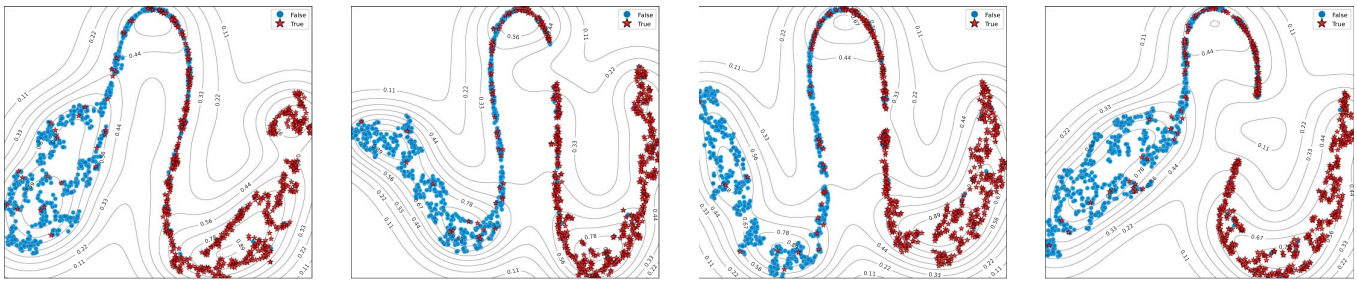
D. Resource Occupancy Analysis in Offline Training

In the experiments, models of different scales exhibited varying resource consumption during data augmentation and fine-tuning processes. For the data augmentation task (Table XVII), both qwen-7B and llama-7B used dual 4090 GPUs with data parallelism, while larger models, qwen-14B and llama-13B, employed model parallelism.



(a) (b) (c) (d)

Fig. 11. Feature distribution of different fine-tuning rounds on Weibo. (a) Proposed-3. (b) Proposed-4. (c) Proposed-5. (d) Proposed-6.



(a) (b) (c) (d)

Fig. 12. Feature distribution of different fine-tuning rounds on PHEME. (a) Proposed-3. (b) Proposed-4. (c) Proposed-5. (d) Proposed-6.

TABLE XVIII
GPU DEVICES AND MEMORY USAGE IN LLM FINE-TUNING

Proposed Methods	Base LLM	Graphics Card Configuration	Average Memory Usage
–4 and –5	Qwen-7B	Dual 4090	20.9GB per card
–8	Llama-7B	Dual 4090	21.8GB per card

Data parallelism involves splitting the data into smaller batches, which are distributed across multiple GPUs. Each GPU processes a subset of the data, and the gradients are averaged across the GPUs during training. This technique is efficient when the model can fit within the memory of a single GPU but requires multiple GPUs to handle larger batches. On the other hand, model parallelism divides the model itself across multiple GPUs. Each GPU stores and processes a portion of the model, allowing for the training of much larger models that exceed the memory capacity of a single GPU. This method is useful for models with large numbers of parameters but comes with higher inter-GPU communication overhead.

The latter models, which require model parallelism, have a significantly higher memory footprint compared to the former models using data parallelism. In the model fine-tuning phase (Table XVIII), both qwen-7B and llama-7B again utilized dual 4090 GPUs, but the memory consumption further increased, highlighting the substantial rise in memory demand during fine-tuning. These results demonstrate that as model size and complexity increase, memory usage grows, with larger models requiring model parallelism for efficient inference, significantly increasing GPU memory consumption compared to smaller models using data parallelism.

V. CONCLUSION

This article introduces LACL, a novel framework that integrates LLMs with CL to enhance misinformation detection in social networks. The innovation lies in leveraging contrastive loss-guided LLM fine-tuning for data augmentation, which generates semantically rich, diverse, and consistent samples to address challenges such as limited labeled data and complex semantics. LACL establishes a feedback loop between the LLM and CL, progressively enhancing both data augmentation and feature extraction. Experimental results on four benchmark datasets demonstrate that LACL consistently outperforms traditional CL-based methods and models that directly apply commercial LLMs without task-specific adaptation. Specifically, LACL achieved significant improvements in detection accuracy and F1-scores, particularly in high-context environments such as Weibo. The study highlights the positive impact of LLM fine-tuning and the number of augmentation rounds on model performance, with fine-tuned models consistently outperforming nonfine-tuned counterparts. Furthermore, the results show that LACL's performance is robust across different LLMs and model sizes, confirming its adaptability to various language settings and its scalability for diverse misinformation detection tasks. Future work will focus on extending LACL to cross-lingual and multimodal settings, while exploring real-time adaptation and hallucination mitigation strategies to enhance robustness and applicability.

REFERENCES

[1] S. A. Aljawarneh and S. A. Swedat, "Fake news detection using enhanced BERT," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 4, pp. 4843–4850, Apr. 2024.

[2] X. Tao, L. Wang, Q. Liu, S. Wu, and L. Wang, "Semantic evolvement enhanced graph autoencoder for rumor detection," in *Proc. ACM Web Conf.*, 2024, pp. 4150–4159.

[3] M.-Y. Chen, Y.-W. Lai, and J.-W. Lian, "Using deep learning models to detect fake news about COVID-19," *ACM Trans. Internet Technol.*, vol. 23, no. 2, 2023.

[4] Y. Liu and Y.-F. B. Wu, "FNED: A deep network for fake news early detection on social media," *ACM Trans. Inf. Syst.*, vol. 38, no. 3, 2020.

[5] S. Dutta, S. Caur, S. Chakrabarti, and T. Chakraborty, "Semi-supervised stance detection of tweets via distant network supervision," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 241–251.

[6] P. Meel and D. K. Vishwakarma, "A temporal ensembling based semi-supervised convnet for the detection of fake news articles," *Expert Syst. Appl.*, vol. 177, 2021, Art. no. 115002.

[7] J. Mukhoti et al., "Open vocabulary semantic segmentation with patch aligned contrastive learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19413–19423.

[8] Y. Liu, L. Huang, F. Giunchiglia, X. Feng, and R. Guan, "Improved graph contrastive learning for short text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 17, 2024, pp. 18716–18724.

[9] S. Zhao, C. Wang, M. Hu, T. Yan, and M. Wang, "MCL: Multi-granularity contrastive learning framework for chinese ner," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 11, 2023, pp. 14011–14019.

[10] S. Li, W. Li, A. M. Luvembe, and W. Tong, "Graph contrastive learning with feature augmentation for rumor detection," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 4, pp. 5158–5167, Nov. 2024.

[11] Y. Jiang, C. Huang, and L. Huang, "Adaptive graph contrastive learning for recommendation," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2023, pp. 4252–4261.

[12] A. Giachanou, P. Rosso, and F. Crestani, "The impact of emotional signals on credibility assessment," *J. Assoc. Inf. Sci. Technol.*, vol. 72, no. 9, pp. 1117–1132, 2021.

[13] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *Proc. Int. Conf. Learn. Representations*, 2020.

[14] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 3438–3445.

[15] Y. Wang, W. Wang, Y. Liang, Y. Cai, J. Liu, and B. Hooi, "NodeAug: Semi-supervised node classification with data augmentation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 207–217.

[16] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," 2019, *arXiv:1901.11196*.

[17] A. Zhong et al., "Logparser-LLM: Advancing efficient log parsing with large language models," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2024, pp. 4559–4570.

[18] B. Wang et al., "Explainable fake news detection with large language model via defense among competing wisdom," in *Proc. ACM Web Conf.*, 2024, pp. 2452–2463.

[19] J. Yang, X. Wang, Y. Zhao, Y. Liu, and F.-Y. Wang, "RAG-based crowdsourcing task decomposition via masked contrastive learning with prompts," *IEEE Trans. Computat. Social Syst.*, vol. 12, no. 4, pp. 1535–1547, Aug. 2025.

[20] L. Huang et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, 2024.

[21] W. Feng, Y. Li, B. Li, Z. Jia, and Z. Chu, "BiMGCL: rumor detection via bi-directional multi-level graph contrastive learning," *PeerJ Comput. Sci.*, vol. 9, 2023, Art. no. e1659.

[22] J. Wu, J. Guo, and B. Hooi, "Fake news in sheep's clothing: Robust fake news detection against LLM-empowered style attacks," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2024, pp. 3367–3378.

[23] F. Liu, X. Zhang, and Q. Liu, "An emotion-aware approach for fake news detection," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 3, pp. 3516–3524, Mar. 2024.

[24] D. Saxena and J. Cao, "Generative adversarial networks (gans) challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–42, 2021.

[25] B. Hu et al., "Bad actor, good advisor: Exploring the role of large language models in fake news detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 20, 2024, pp. 22105–22113.

[26] Y. Li et al., "CALRec: Contrastive alignment of generative LLMs for sequential recommendation," in *Proc. 18th ACM Conf. Recommender Syst.*, 2024, pp. 422–432.

[27] Q. Dong et al., "Unsupervised large language model alignment for information retrieval via contrastive feedback," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2024, pp. 48–58.

[28] C. Jiang et al., "Hallucination augmented contrastive learning for multimodal large language model," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit.*, 2024, pp. 27036–27046.

[29] B. Chen and A. L. Bertozzi, "AutoKG: Efficient automated knowledge graph generation for language models," in *Proc. IEEE Int. Conf. Big Data*, 2023, pp. 3117–3126.

[30] B. Xie et al., "Multiknowledge and llm-inspired heterogeneous graph neural network for fake news detection," *IEEE Trans. Computat. Social Syst.*, vol. 12, no. 2, pp. 682–694, Apr. 2025.

[31] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3651–3657.

[32] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Circuits Syst.*, vol. 8, no. 7, pp. 579–588, 2009.

[33] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2006, pp. 1735–1742.

[34] K. Lv, Y. Yang, T. Liu, Q. Gao, Q. Guo, and X. Qiu, "Full parameter fine-tuning for large language models with limited resources," 2023, *arXiv2306.09782*.

[35] Y. Gu, X. Han, Z. Liu, and M. Huang, "PPT: Pre-trained prompt tuning for few-shot learning," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 8410–8423.

[36] N. Ding et al., "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Mach. Intell.*, vol. 5, no. 3, pp. 220–235, 2023.

[37] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," 2021, *arXiv2106.09685*.

[38] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[39] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 708–717.

[40] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, "Jointly embedding the local and global relations of heterogeneous graph for rumor detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM),* 2019, pp. 796–805.

[41] E. Kochkina, M. Liakata, and A. Zubiaga, "All-in-one: Multi-task learning for rumour verification," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3402–3413.

[42] T. Bian et al., "Rumor detection on social media with bi-directional graph convolutional networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 549–556.

[43] H. Aydın, Z. Orman, and M. A. Aydın, "A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment," *Comput. Secur.*, vol. 118, 2022, Art. no. 102725.

[44] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods in Natural Lang. Process. (EMNLP),* Doha, Qatar, 2014, pp. 1746–1751.

[45] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[46] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics:* Human Lang. Technol., 2016, pp. 1480–1489.

[47] Y. Yan, P. Zheng, and Y. Wang, "Enhancing large language model capabilities for rumor detection with knowledge-powered prompting," *Eng. Appl. Artif. Intell.*, vol. 133, 2024, Art. no. 108259.

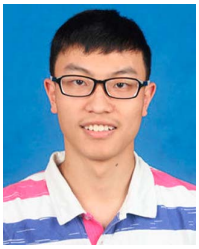
[48] Y. Gao, X. Wang, X. He, H. Feng, and Y. Zhang, "Rumor detection with self-supervised learning on texts and social graph," *Frontiers Computer Sci.*, vol. 17, no. 4, 2023, Art. no. 174611.

[49] J. Bai et al., "Qwen technical report," 2023, *arXiv:2309.16609*.
[50] H. Touvron et al., "Llama: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

Hang Shen (Member, IEEE) received the Ph.D.(Hons.) degree in computer science from Nanjing University of Science and Technology, Nanjing, China, in 2015.

From 2018 to 2019, he worked as a Full-Time Postdoctoral Fellow with the Broadband Communications Research (BBCR) Laboratory, Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the Department of Computer Science and Technology, Nanjing Tech University, Nanjing. His research interests include wireless networking and cybersecurity.

Dr. Shen serves as an Associate Editor for the *Journal of Information Processing Systems*, *Frontiers in Blockchain*, and IEEE ACCESS, and a Guest Editor for *Peer-to-Peer Networking and Applications*. He was a Program Committee Member of the 2025 International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), the 2024 IEEE International Conference on High Performance Computing and Communications (HPCC), and the 2021 Annual International Conference on Privacy, Security and Trust (PST). He is an Executive Committee Member of the ACM Nanjing Chapter and a Supervisory Committee Member of the CCF Nanjing Chapter.



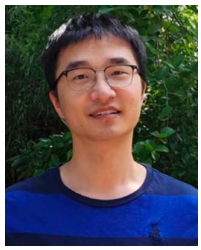
Xiang Li received the B.Eng. degree in network engineering from Nanjing Institute of Technology, Nanjing, China, in 2022. He is currently working toward the M.Eng. degree in computer science with Nanjing Tech University, Nanjing.

His research interests include large language models and knowledge graphs in the context of cybersecurity.

He is a CCF Student Member.

Xu Wang received the B.Sc. degree in information and computational science and the M.Eng. degree in computer science from Nanjing Tech University, Nanjing, China, in 2022 and 2025.

He is currently an Engineer at Zhenjiang Telecom, Zhenjiang, China. His research interests include unsupervised and semisupervised learning, as well as large language models in the context of cybersecurity.



Yuanfei Dai received the Ph.D. degree in computer science from Fuzhou University, Fuzhou, China, in 2021.

From 2019 to 2020, he was a Visiting Student with Brown University, Providence, RI, USA. He is currently a Lecturer with Nanjing Tech University, Nanjing, China. His research interests include knowledge acquisition and knowledge-graph representation.

Dr. Dai, together with his student, received the Best Student Paper Award at the 2023 International Conference on Knowledge Science, Engineering and Management.

Tianjing Wang (Member, IEEE) received the B.Sc. degree in mathematics from Nanjing Normal University, Nanjing, China, in 2000, the M.Sc. degree in mathematics from Nanjing University, Nanjing, in 2002, and the Ph.D. degree in signal and information system from Nanjing University of Posts and Telecommunications (NUPT), Nanjing, in 2009.

She is an Associate Professor with Nanjing Tech University, Nanjing. From 2011 to 2013, she was a Full-Time Postdoctoral Fellow with the School of Electronic Science and Engineering, NUPT. From 2013 to 2014, she served as a Visiting Scholar with the Electrical and Computer Engineering Department, State University of New York, Stony Brook, New York, USA. Her research interests include cybersecurity and vehicular networks.

Guangwei Bai received the B.Eng. and M.Eng. degrees in computer engineering from Xi'an Jiaotong University, Xi'an, China, in 1983 and 1986, respectively, and the Ph.D. degree in computer science from the University of Hamburg, Hamburg, Germany, in 1999.

From 1999 to 2001, he worked as a Research Scientist with the German National Research Center for Information Technology, Bonn, Germany. In 2001, he joined the University of Calgary, Calgary, AB, Canada, as a Research Associate. Since 2005, he has been working with Nanjing Tech University, Nanjing, China, as a Professor in computer science. From 2010, he was a Visiting Professor with the Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, Canada. His research interests include wireless networking and cybersecurity.