Cross-domain Rumor Detection via Test-Time Adaptation and Large Language Models

Yuxia Gong, Shuguo Hu, Huaiwen Zhang*
College of Computer Science, Inner Mongolia University
yuxiagong@mail.imu.edu.cn, shuguo.hu@mail.imu.edu.cn, huaiwen.zhang@imu.edu.cn

Abstract

Rumor detection on social media has become crucial due to the rapid spread of misinformation. Existing approaches primarily focus on within-domain tasks, resulting in suboptimal performance in cross-domain scenarios due to domain shift. To address this limitation, we draw inspiration from the strong generalization capabilities of Test-Time Adaptation (TTA) and propose a novel framework to enhance rumor detection performance across different domains. Specifically, we introduce **T**est-**T**ime **A**daptation for **R**umor **D**etection (**T^{2}ARD**), which incorporates both single-domain model and target graph adaptation strategies tailored to the unique requirements of cross-domain rumor detection. T^{2}ARD utilizes a graph adaptation module that updates the graph structure and node attributes through multi-level self-supervised contrastive learning, aiming to derive invariant graph representations. To mitigate the impact of significant distribution shifts on self-supervised signals, T^{2}ARD performs model adaptation by using annotations from Large Language Models (LLMs) on target graph to produce pseudo-labels as supervised signals. Experiments conducted on four widely used cross-domain datasets demonstrate that T^{2}ARD achieves state-of-the-art performance, surpassing existing methods in rumor detection.

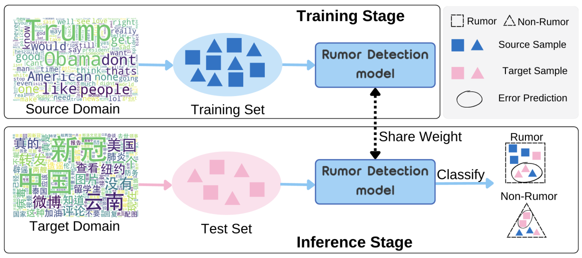


Figure 1: An illustration example of cross-domain rumor detection. The rumor detection models trained on the source domain may not adapt well to the target domain. More error-classifying results are observed when directly inferring the target domain samples on the model.

1 Introduction

With the advancement of mobile Internet technology (Han and Li, 2021), social networks such as Weibo and X (Twitter) have become key channels for information dissemination (Qalati et al., 2023). While these platforms enable rapid information sharing and diverse opinions, they also provide a fertile ground for rumor proliferation, especially during emergent events. Due to the vast user base and ease of access, rumors can propagate rapidly (Zubiaga et al., 2018), causing significant societal disruptions (Chen and Wang, 2020). Therefore, there is an urgent need for effective strategies for rumor detection on social media.

Recent approaches (Ma et al., 2018; Bian et al., 2020; Tian et al., 2022; Sun et al., 2022; Zhang et al., 2021; Zhu et al., 2024) have modeled rumor propagation as graphs to capture structural features. While these models have achieved success, most methods (Bian et al., 2020; Tian et al., 2022; Zhu et al., 2024) assume that test data follow the same distribution as training data, which is relatively less the case in real-world scenarios. In practice, models trained on source domains often suffer significant performance degradation when deployed to target domains due to domain discrepancies (Lin et al., 2022, 2023; Tang et al., 2023; Liu et al., 2024; Ding et al., 2025). As shown in Figure 1, rumor detection models trained on a source domain often perform poorly when applied to a target domain. This occurs because these models are trained on data from limited platforms but are expected to generalize across a variety of platforms. Similarly, training data are often limited to specific topics or events within fixed time periods, while models must handle unseen topics and adapt to dynamic environments. These distributional mismatches be-

*Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 8051–8066
November 4-9, 2025 ©2025 Association for Computational Linguistics

tween training and testing data lead to suboptimal detection performance. To address this issue, prior work (Jin et al., 2022) explored adapting and refining graph data during the test phase, without relying on labeled source graph data, to improve model performance under distribution shifts.

In this paper, we explore efficient test-time adaptation methods to reduce the distribution shift between source and target domains in cross-domain rumor detection. The essence of TTA is to improve a model's generalization ability on specific test data. Traditional TTA methods (Jin et al., 2022; Wang et al., 2022; Chen et al., 2022; Zhang et al., 2024c) rely on self-supervised signals to modify test data during the adaptation phase, enabling adaptation to unseen domains. However, these methods face two critical challenges: (1) How to design effective self-supervised signals for rumor detection task? While self-supervised signals optimize graph structures and node features, they may not align with downstream task objectives, leading to over-optimization and reduced downstream performance. (2) How to assist self-supervised tasks under significant distribution shifts between the source and target domains? Self-supervised signals assume some similarity between source and target distributions. However, large shifts can hinder the effectiveness of these signals, leading to performance decline. Recent studies (Gui et al., 2024) suggest that incorporating a few labeled test instances can improve performance across test domains with theoretical guarantees. Building on this, we propose integrating a few supervised signals at test time to further enhance model performance.

To address the aforementioned challenges, we propose a novel **T**est-**T**ime **A**daptation framework for **R**umor **D**etection (**T^{2}ARD**), aimed at enhancing the efficacy of rumor detection in cross-domain scenarios. Specifically, to tackle the first challenge, T^{2}ARD performs *graph-view* adaptation by leveraging graph structures to represent social media conversations and employing a multi-level self-supervised contrastive learning approach to guide the refinement of graph data, thereby facilitating test-time adaptation on graph data. For the second challenge, inspired by the significant advancements of Large Language Models, T^{2}ARD conducts *model-view* adaptation by incorporating LLMs as annotators to assign pseudo-labels to target graph data. Based on the prediction confidence of the LLMs, the framework filters graphs for supervised learning, enabling more effective adaptation during the test phase. Empirical evaluations conducted on various benchmarks demonstrate the superior performance of our proposed method compared to state-of-the-art approaches. Our contributions are summarized as follows:

- We investigate the novel task of test-time adaptation for rumor detection, enabling models to retain training knowledge while effectively adapting to the unique characteristics of test samples.
- We propose the T^{2}ARD method, which utilizes multi-level self-supervised learning tasks to enhance model generalization and leverages LLMs as annotators to generate pseudo-labels to adapt pre-trained rumor detectors at test time.
- Extensive experiments conducted on four widely used cross-domain datasets demonstrate the effectiveness of our proposed method.

2 Related Work

2.1 Rumor Detection

Most existing rumor detection methods focus on in-domain data and build various frameworks tailored to that setting. For example, sequence-based models leverage the textual content of the original posts and user response comments for rumor detection (Ma et al., 2016, 2019, 2021), propagation structure-aware methods (Bian et al., 2020; Zhang et al., 2021; Wu and Hooi, 2023; Cui and Jia, 2024, 2025) model the propagation paths as tree-structured graphs enriched with the textual content to build the semantics of posts and their propagation relationships. However, in real-world scenarios, events continually emerge from previously unseen domains. Existing rumor detection models face a significant challenge since there is no sufficient available labeled data in target domains. To tackle this challenge, recent works proposed contrastive learning models (Lin et al., 2022, 2024; Cui and Jia, 2025), zero-shot response-aware prompt Learning method (Lin et al., 2023) and test-time training for rumor detection (Zhang et al., 2024a; Tao et al., 2024) to improve cross-domain generalization. However, employing these methods in practice may be infeasible, as they rely on labeled source graph data and incur additional costs associated with modifying model architectures or re-training model parameters. Moreover, we note that T^{2}ARD has an essential difference from (Tao et al., 2024). Our work focuses on TTA while (Tao et al., 2024) addresses Test-time training (TTT).

2.2 Test-Time Adaptation

Test Time Adaptation (Liang et al., 2024) is a paradigm that improves a model's generalization to target test data by performing unsupervised fine-tuning at inference time. Note that the model has been trained on a separate training dataset, which is not available during the adaptation phase. Test Time Training (TTT) (Sun et al., 2020; Wang et al., 2022; Zhang et al., 2024b) adapt models at test time via a self-supervised auxiliary task but require the training of the same auxiliary task during training process. BN (Schneider et al., 2020) updates the BatchNorm (Ioffe, 2015) statistics using test data. TENT (Wang et al., 2020) extends this idea by minimizing entropy to adapt BatchNorm layers. SHOT (Liang et al., 2020) combines entropy minimization with pseudo-labeling. EATA (Niu et al., 2022) achieves efficient adaptation by selectively updating samples with Fisher regularization. Besides these applications, TTA has also been investigate in the graph domain (Jin et al., 2022; Mao et al., 2024; Bao et al., 2024; Zhang et al., 2024d). For instance, GTrans (Jin et al., 2022) transforms the test graph to enhance generalization while leaving the pre-trained model fixed. In this work, we propose a novel TTA method tailored for rumor detection task.

2.3 LLMs for Graphs

Large language models demonstrate strong zero-shot and few-shot capabilities owing to their massive parametric knowledge. Considerable research (Guo et al., 2023; He et al., 2023; Yu et al., 2025) has increasingly explored applying LLMs to graph-related tasks. Relying solely on LLMs as predictors (Ye et al., 2023; Chen et al., 2024; Wang et al., 2024) constitutes a promising strategy, with GPT4Graph (Guo et al., 2023) evaluating the potential of LLMs for graph classification tasks. NLGraph (Wang et al., 2024) proposes a comprehensive benchmark for evaluating graph-structure reasoning abilities. In this paper, we leverage LLMs as annotators to train an efficient model independent of any ground-truth labels.

3 Methodology

3.1 Preliminary and Problem Definition

We mainly focus on the rumor detection task in the cross-domain setting. In this work, the cross-domain rumor detection task is formulated as follows: given a source dataset, classify each event in a distinct target dataset as a rumor or non-rumor, where the source and target data are from different domains. Specifically, we define a source dataset for training as $\mathcal{D}^s = \{C_1^s, C_2^s, \cdots, C_N^s\}$, where N denotes the number of source events. Each source event $C^s = \{y, c, \mathcal{R}(c)\}$ consists of the source claim c, its responsive posts $\mathcal{R}(c)$ and a label $y \in \{\text{rumor}, \text{non-rumor}\}$ indicating whether C^s is a rumor. The responsive posts $\mathcal{R}(c) = \{c, r_1^s, r_2^s, \cdots, r_n^s\}$, where r_i^s is the i-th responsive post text, and n is the total number of responsive posts. For evaluation, we consider the target dataset with a different language and domain from the source dataset as $\mathcal{D}^t = \{C_1^t, C_2^t, \cdots, C_M^t\}$, where M is the number of target events. Each target event $C^t = \{c', \mathcal{R}(c')\}$ shares the similar structure as that of the source.

Given an event C, we construct the propagation graph as $\mathrm{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V}$ and $\mathcal{E}$ represent the sets of node and edge. The connectivity is captured by an adjacent matrix $\mathrm{A} \in \mathbb{R}^{n\times n}$ and $\mathrm{A}_{i,j} = 1$ if v_i connects to v_j, while the node feature matrix $\mathrm{X} \in \mathbb{R}^{n\times d}$ contains the node feature representations. In this notation, n corresponds to the number of nodes and d indicates the dimension of the node features. Our goal is to infer the label of G^t in the target domain using a model $f_\theta(\mathrm{G}^t) \rightarrow y^t$ trained on $\mathcal{D}^s$.

We tackle the aforementioned challenges by jointly optimizing the target graph data and the pre-trained rumor detection model to narrow the gap between the source and target domains. Figure 2 illustrated an overview of our proposed T^{2}ARD, which consists of two core components: the graph adaptation module and the model adaptation module.

3.2 The Test-time Adaptation Framework

T^{2}ARD is provided in the Figure 2 that includes three parts: the pre-training phase, the test-time adaptation phase, and the inference phase as follows:

Pre-training phase. This phase aims to obtain a pre-trained rumor detection model with optimized parameters capable of accurately predicting labels

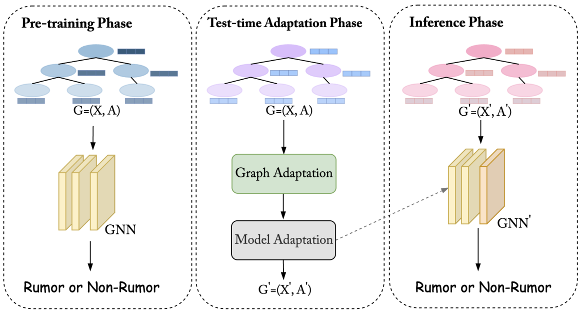


Figure 2: The overall framework of T^{2}ARD.

on the source training data $\mathcal{D}^s$. Notably, the subsequent test-time adaptation phase relies solely on the model parameters $\boldsymbol{\theta}^*$ and the test data $\mathcal{D}^t$, thereby rendering T^{2}ARD model-agnostic.

The Test-time Adaptation phase. Our approach employs test-time adaptation to handle the cross-domain problem on rumor detection. We encounter several challenges in achieving this goal: (1) How to design effective self-supervised signal for rumor detection task? To address this issue, T^{2}ARD proposes a graph adaptation module with self-supervised contrastive learning integrating knowledge from the pre-trained rumor detection model with the test graphs characteristics. Detailed illustration is provided in Section 3.3. (2) How to assist self-supervised tasks under significant distribution shifts between the source and target domains? To tackle this challenge, we propose a prompt-based model adaptation module that utilizes LLMs to generate confidence-aware annotations. Additional information is available in Section 3.4.

Inference phase. During the inference stage, the refined test data together with the updated rumor detection model are employed to predict labels for the test set.

3.3 Graph Adaptation With Self-supervised Contrastive Learning

Graph Feature and Structure Refinement. We introduce two basic transformation functions: $\mathrm{X}' = \sigma(\mathrm{X})$ which derives new features by adding or masking values in X, and $\mathrm{A}' = \psi(\mathrm{A})$ which yields a new adjacent matrix by inserting or removing edges in A. The self-supervised graph adaptation method aims to learn rumor detection-specific optimal functions that mitigate domain shift. However, the task is challenging due to the absence of supervision and the unavailability of source graph data. Therefore, we adopt two extremely straightforward strategies below.

Given node feature matrix X, Equation 1 defines a feature transformation that enhances the node features via an additive function. Here, $\Delta\mathrm{X} \in \mathbb{R}^{n\times d}$ serves as continuous free parameters, offering substantial flexibility. This scheme supports either masking node features with zeros or modifying them with alternate values. Similarly, the graph topology is adjusted as in equation 2, where $\Delta\mathrm{A} \in \mathbb{R}^{n\times n}$ represents a binary matrix to refine each node's neighborhood and $\oplus$ denotes the element-wise exclusive XOR operation. Specifically, if the corresponding entries in A and ΔA are both 1, the XOR operation returns 0, resulting in deletion of the corresponding edge. If the corresponding entries in A and ΔA are 0 and 1 respectively, this results in an edge additions. To avoid large deviations from the original graph structure, we constrain the number of modified entries in the adjacency matrix to be at most a predetermined budget $\mathcal{B}$, i.e., $\sum \Delta\mathrm{A} \leq \mathcal{B}$, which narrows the search space and improves computational efficiency.

$$\mathrm{X}' = \sigma(\mathrm{X}) = \mathrm{X} + \Delta\mathrm{X} \tag{1}$$

$$\mathrm{A}' = \psi(\mathrm{A}) = \mathrm{A} \oplus \Delta\mathrm{A} \tag{2}$$

Self-supervised Contrastive Learning. To optimize the free-parameters ΔX and ΔA, we adopt a Self-supervised Contrastive Learning objective to guide the graph adaptation procedure, since ground-truth labels are unavailable in this setting. Specifically, this mechanism comprises two self-supervised contrastive learning tasks, namely, macro- and micro-level contrastive learning, to capture the intrinsic characteristics of social media conversations.

The objective of Macro-level Contrastive Learning (MacroCL) is to learn domain-invariant global features by assisting node representations with the structural information of the entire social media conversation graph. In essence, MacroCL maximizes the mutual information between the node-level representations and macro-level graph representation. As illustrated in Figure 3, starting from the original rumor propagation graph (purple), we generate four different views (orange) via data augmentation. For MacroCL, we employ two views: the original view G_0, where the structure and attribute of graph remains unaltered; the shuffled node attributes view G_1, where the structure of graph remains unaltered, while the attributes of nodes are randomly shuffled. The corresponding node representations, X_0 and X_1,

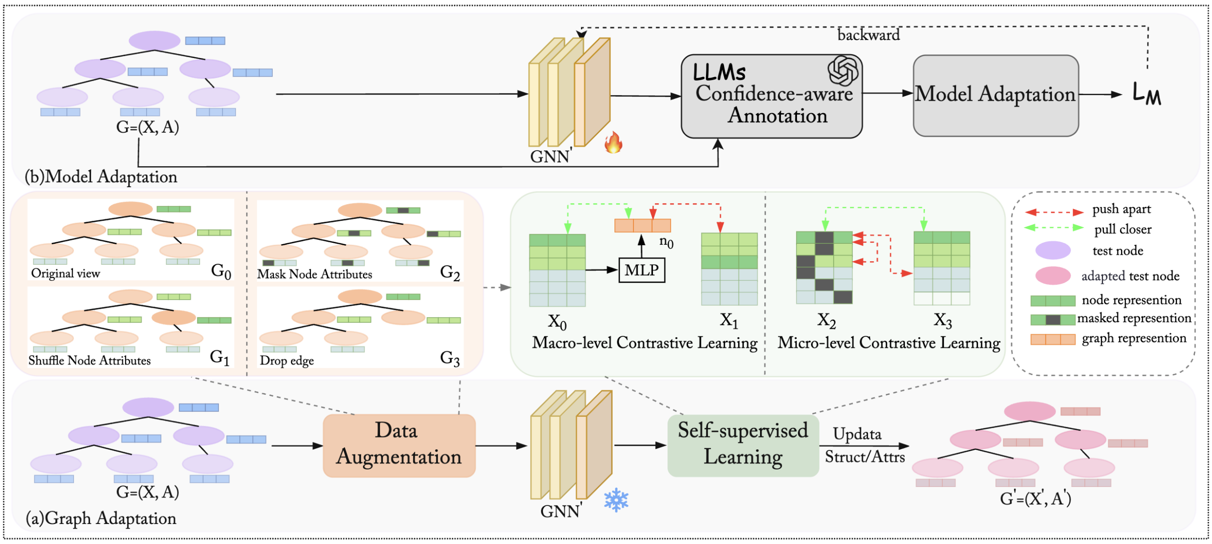


Figure 3: The architecture of our proposed T²ARD in the Test-Time Adaptation Phase: graph adaptation and model adaptation.

are derived by feeding the two views into the pre-trained rumor detection model. Following DGI (Veličković et al., 2018), a Macro-level graph representation is obtained via $\mathrm{n}_0 = \mathrm{MLP}(\mathrm{X}_0)$, where the node representation matrix X_0 is extracted from the original view G_0. The positive pairs of MacroCL $\{\langle \mathrm{x}_{0,j}, \mathrm{n}_0 \rangle\}_{j=1}^{|n|}$, couple the node representation $\mathrm{x}_{0,j}$ of original view G_0 with the graph representation n_0 of view G_0, while the negative pairs $\{\langle \mathrm{x}_{1,j}, \mathrm{n}_0 \rangle\}_{j=1}^{|n|}$ are composed of the node representation $\mathrm{x}_{1,j}$ from the view G_1 and the graph representation n_0. The distance between node and graph representation is measured as $d(\mathrm{x}_{i,j}, \mathrm{n}_i) = \mathrm{Sigmoid}(\mathrm{x}_{i,j} * \mathrm{n}_i)$, where $*$ denotes the inner product. To ensure higher scores for positive pairs and lower scores for negative pairs, we define the objective function for MacroCL is delineated as follows:

$$L_{\mathrm{MacroCL}} = -\frac{1}{2|n|} \sum_{j=1}^{|n|} \left(\log d(\mathrm{x}_{0,j}, \mathrm{n}_0) + \log(1 - d(\mathrm{x}_{1,j}, \mathbf{n}_0)) \right) \tag{3}$$

We further introduce Micro-level Contrastive Learning (MicroCL) task aimed at learning a robust node representation which able to tolerate slight perturbations of attributes and structures within social media conversations. Specifically, MicroCL aims to learn domain-invariant local features by modeling fine-grained node characteristics, ensuring robustness to minor perturbations in both attributes and structural patterns. Given a rumor propagation structure input graph G (purple), as shown in Figure 3, four views (orange) of the graph can be generated through data augmentation. Specifically, we mask some attributes of nodes in view G_2, and randomly drop some edges in G_3; neither operation introduces significant alterations to the input graph. Feeding the view G_2 and G_3 into the pre-trained rumor detection model yields the corresponding node representations X_2 and X_3, respectively. The positive pairs of MicroCL $\{\langle \mathrm{x}_{2,j}, \mathrm{x}_{3,j} \rangle\}_{j=1}^{|n|}$ consist of the same nodes in two augmented views. The negative pairs are $\{\langle \mathrm{x}_{2,i}, \mathrm{x}_{3,j} \rangle\}_{i \neq j}^{|n|}$ and $\{\langle \mathrm{x}_{2,i}, \mathrm{x}_{2,j} \rangle\}_{i \neq j}^{|n|}$. Inspired by InfoNCE (Oord et al., 2018; Zhu et al., 2021), the objective for a positive node pair $(\mathrm{x}_{2,i}, \mathrm{x}_{3,i})$ is defined as follows:

$$I_{\mathrm{MicroCL}}(\mathrm{x}_{2,i}, \mathrm{x}_{3,i}) = \log \frac{\exp\left(\frac{\mathrm{F}_{2,i}^{\top} \mathrm{F}_{3,i}}{\tau}\right)}{\sum_{j \neq i} \left(\exp\left(\frac{\mathrm{F}_{2,i}^{\top} \mathrm{F}_{3,j}}{\tau}\right) + \exp\left(\frac{\mathrm{F}_{2,i}^{\top} \mathrm{F}_{2,j}}{\tau}\right) \right)} \tag{4}$$

where $\mathrm{F}_{ij} = \mathrm{MLP}(\mathrm{x}_{ij})$ and MLP is a two-layer perceptron. The objective function for MicroCL can be defined as follows:

$$L_{\mathrm{MicroCL}} = -\frac{1}{2|n|} \sum_{i=1}^{|n|} \left(I_{\mathrm{MicroCL}}(\mathrm{x}_{2,i}, \mathrm{x}_{3,i}) + I_{\mathrm{MicroCL}}(\mathrm{x}_{3,i}, \mathrm{x}_{2,i}) \right) \tag{5}$$

The total loss function of Self-supervised Contrastive Learning is a weighted combination of MacroCL and MicroCL losses:

$$L_{\mathrm{G}} = L_{\mathrm{MacroCL}} + \alpha L_{\mathrm{MicroCL}} \tag{6}$$

where α is the parameter that balances macro- and micro-level contrastive learning.

3.4 Prompt-based Model Adaptation With LLMs

As evidenced by the empirical results in Section 4.3, the quality of the self-supervised signal is critical for the performance of test-time adaptation . However, to reduce the impact of distribution shifts, the proposed T^{2}ARD creatively suggests using pseudo-labels for semi-supervised adaptation at test time.

Confidence-aware Annotation. Inspired by existing work on leveraging LLMs for graphs (Chen et al., 2023; He et al., 2023), T^{2}ARD proposes prompting LLMs to generate pseudo-labels. Specifically, the prompt incorporates prediction results from the pre-trained rumor detection model. Moreover, to assess the quality of LLM's annotations, we additionally request the prediction confidence score for the LLM-generated pseudo-labels.

Model Adaptation. The pseudo labels produced by LLMs may be noisy and consequently influence the model. Therefore, we obtain the confidence of the LLMs' prediction through LLMs as Annotators for Pseudo-Labels. To mitigate the potential impact of noisy pseudo-labels, T^{2}ARD performs graph filtering in each batch by excluding graphs based on confidence scores. Graphs annotated with high confidence score are chosen for the test-time learning. Then the filtered graphs and their corresponding pseudo-labels are utilized as supervision for model adaption. The predicted label $y = f_{\theta_m}(\text{G'})$ is obtained by inputting the transformed graph G' $=$ (X', A') into several full connection layers with parameter $\boldsymbol{\theta_m}$ and a softmax function. Given the filtered graphs, we fine-tune the model by minimizing the cross-entropy loss between the prediction y and the pseudo-labels $\hat{y}$:

$$L_M(G', \hat{y}; \boldsymbol{\theta_m}) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(g_\theta(G'); \boldsymbol{\theta}^*)\right) \quad (7)$$

where N is the number of filtered graphs and $\boldsymbol{\theta}^*$ is the pre-trained GNN parameters.

4 Experiment

4.1 Experimental Settings

Datasets. We evaluate the proposed model on four real-world cross-domain rumor sets: (i) English TWITTER and Twitter-COVID-19, (ii) English TWITTER and Chinese Weibo-COVID-19, (iii) Chinese WEIBO and Weibo-COVID-19, and (iv) Chinese WEIBO and English Twitter-COVID-19. These cross-domain datasets are annotated with two binary labels: Non-rumor and Rumor. For detailed statistics and descriptions of the datasets, please refer to Appendix A.

Baselines. We compare T^{2}ARD with three categories of baselines: traditional rumor detection approaches, LLM-based approaches, and cross-domain rumor detection approaches. Traditional category contains: 1) CNN (Yu et al., 2017); 2) RNN (Ma et al., 2016); 3) RvNN (Ma et al., 2018); 4) PLAN (Khoo et al., 2020); 5) BiGCN (Bian et al., 2020). LLM-based category contains: 6) Llama3.1-8B; 7) Qwen2.5-72B (Yang et al., 2024); 8) GPT-4o. And cross-domain category contains: 9) ACLR-BiGCN (Lin et al., 2022); 10) RPL (Lin et al., 2023); 11) T^{3}RD (Zhang et al., 2024a). This work focuses on the most challenging scenario: detecting target events in a unseen domain. Specifically, TWITTER and WEIBO serve as the source datasets, whereas Twitter-COVID19 and Weibo-COVID19 are treated as the target datasets.

Evaluation and Implementation. We employ the commonly used metric, i.e., Accuracy (Acc) to evaluate model performance. The F1-score is reported separately for the positive class (RF_1), the negative class (NF_1), and the macro-average (Mac-F_1). Gpt-4o-2024-05-13 is adopted to generate pseudo-labels. The prompting strategy for generating pseudo-labels is detailed in Appendix B. And more implementation details are shown in Appendix C.

4.2 Overall Performance

Table 1 presents the performance of the proposed method alongside all baselines across the four sets of cross-domain datasets, with the best performances highlighted in bold. From Table 1, the first set of experiments is based on the within-domain rumor detection methods, and the second set of results is based on the LLMs approaches. We can observe that the results of the first set of experiments underperform those of the second set of experiments, since LLMs are pre-trained on massive, diverse corpora that let them capture some common features of cross-domain rumors even without fine-tuning. Among the LLM-based baselines in the second set, the performance is worse than dedicated cross-domain rumor detection methods, as these methods rely solely on general semantic understanding for inference and lack task-specific decision boundaries. The third set focuses on cross-domain rumor detection, with ACLR aligning the source and target domains through supervised contrast learning, RPL investigates efficient prompting

Source	TWITTER								WEIBO							
Target	Twitter-COVID19				Weibo-COVID19				Weibo-COVID19				Twitter-COVID19			
Model	Acc.	Mac-F_1	RF_1	NF_1	Acc.	Mac-F_1	RF_1	NF_1	Acc.	Mac-F_1	RF_1	NF_1	Acc.	Mac-F_1	RF_1	NF_1
CNN	0.406	0.366	0.450	0.285	0.429	0.415	0.441	0.389	0.421	0.410	0.438	0.382	0.415	0.360	0.432	0.288
RNN	0.419	0.394	0.431	0.357	0.451	0.431	0.469	0.393	0.432	0.427	0.458	0.396	0.427	0.368	0.442	0.293
RvNN	0.436	0.430	0.458	0.401	0.479	0.410	0.437	0.383	0.471	0.493	0.548	0.437	0.432	0.421	0.451	0.391
PLAN	0.455	0.454	0.432	0.476	0.385	0.384	0.301	0.466	0.384	0.372	0.283	0.461	0.462	0.455	0.432	0.477
BiGCN	0.624	0.566	0.729	0.402	0.615	0.524	0.729	0.319	0.612	0.561	0.681	0.441	0.545	0.529	0.511	0.547
Llama3.1-8B	0.496	0.432	0.448	0.511	0.438	0.417	0.407	0.473	0.403	0.392	0.484	0.357	0.372	0.331	0.401	0.332
Qwen2.5-72B	0.536	0.507	0.485	0.570	0.519	0.486	0.451	0.554	0.455	0.413	0.527	0.432	0.427	0.391	0.472	0.381
GPT-4o	0.643	0.629	0.598	0.681	0.621	0.607	0.560	0.681	0.627	0.566	0.695	0.477	0.581	0.550	0.602	0.553
ACLR-BiGCN	0.759	0.710	0.808	0.612	0.721	0.685	0.788	0.582	0.695	0.671	0.756	0.585	0.676	0.642	0.739	0.545
RPL	0.780	0.739	0.823	0.654	0.745	0.719	0.804	0.634	0.734	0.717	0.822	0.612	0.727	0.697	0.793	0.601
T^{3}RD	0.823	0.803	**0.833**	0.773	0.797	0.788	**0.832**	0.743	0.751	0.715	0.828	0.602	0.735	**0.701**	**0.808**	0.593
T^{2}ARD	**0.854**	**0.846**	0.806	**0.886**	**0.833**	**0.811**	0.746	**0.876**	**0.812**	**0.789**	**0.857**	**0.720**	**0.781**	0.696	0.706	**0.685**

Table 1: The experimental results on the Target domain.

with language and domain transfer for zero-shot rumor detection and T^{3}RD employs test-time training to further extract additional information from the test data.

In contrast, our proposed T^{2}ARD approach achieves state-of-the-art performance among all baselines, indicating strong generalization for cross-domain transfer between different domains. In comparison with the previous state-of-the-art, T^{2}ARD achieves a 6.1% improvement in accuracy score on the third set of datasets. On the first set dataset, T^{2}ARD also exhibits competitive performance, with a 3.1% improvement in accuracy over the previous state-of-the-art T^{3}RD model. The generalization of the T^{2}ARD under substantial distribution shifts is further substantiated by the results on the second and fourth groups. Our proposed T^{2}ARD achieves the best performance in Twitter-COVID19 dataset, with a 4.6% improvement in accuracy over the T^{3}RD model. These results further underscore the efficacy of the test-time adaptation in mitigating the distribution shift for cross-domain rumor detection by graph adaptation and model adaptation.

Moreover, while T^{2}ARD significantly improves overall ACC, it also inevitably misfilters some low-confidence true rumor samples in the model adaptation module. Since rumors often contain exaggerated or unverified information that is not adequately represented in the LLM knowledge base, resulting lower confidence scores and consequently a slightly reduced RF_1 compared to T^{3}RD. Case studies are provided in Appendix D. And further algorithmic complexity analysis appears in Appendix E.

Source	TWITTER		WEIBO	
Model	Acc.	Mac-F$_1$	Acc.	Mac-F$_1$
T^{2}ARD	0.833	0.811	0.812	0.789
T^{2}ARD w/o TTA	0.615	0.524	0.612	0.561
T^{2}ARD w/o L_G	0.646	0.581	0.651	0.584
T^{2}ARD w/o $L_{MacroCL}$	0.766	0.752	0.744	0.723
T^{2}ARD w/o $L_{MicroCL}$	0.682	0.641	0.680	0.670
T^{2}ARD w/o L_M	0.802	0.789	0.773	0.761
T^{2}ARD w/o $conf$	0.832	0.821	0.762	0.702

Table 2: Ablation studies on our proposed model.

4.3 Ablation Study

Ablation analysis is performed on the Weibo-COVID19 dataset, achieved by discarding some important components of our best performed approach T^{2}ARD.

The Effect of Test-time Adaptation. Table 2 presents an ablation analysis evaluating the effectiveness of test-time adaptation on rumor detection. Specifically, we derive the T^{2}ARD-w/o-TTA variant by removing the test-time adaptation phase from T^{2}ARD. As depicted in Table 2, performance reveal a gradual decline, underscoring the crucial role of test-time adaptation in mitigating the distribution shift between source and target domain for cross-domain rumor detection.

The Effect of components in T^{2}ARD. To investigate the contribution of each component in T^{2}ARD, we show the effectiveness of our proposed adaptation mechanism in Table 2. Specifically, the pre-trained rumor detection model is denoted as BiGCN and we strengthen the pre-trained rumor detection model with graph adaptation (L_G), model adaptation (L_M), Macro-level Contrastive Learning ($L_{MacroCL}$), Micro-level Contrastive Learning ($L_{MicroCL}$) and confidence-aware annotation

Source	TWITTER		WEIBO	
Model	Acc.	Mac-F_1	Acc.	Mac-F_1
PMRD	0.615	0.524	0.612	0.561
PMRD + Entropy	0.720	0.706	0.665	0.650
PMRD + SLAPS	0.757	0.743	0.696	0.620
PMRD + L_G	0.802	0.789	0.773	0.761
T^2ARD	**0.833**	**0.811**	**0.812**	**0.789**

Table 3: Performance with different graph adaptation strategies.

($conf$) respectively. As can be observed, both graph adaptation and model adaptation modules improve the performance of pre-trained rumor detection model, but the graph adaptation module plays a more significant role than the model adaptation module. This advantage arises because the graph adaptation module often captures more transferable knowledge for cross-domain rumor detection through self-supervised learning, whereas model adaptation module might be less crucial when the underlying pseudo-labels are not always accurate. Moreover, Macro-level Contrastive Learning ($L_{MacroCL}$) and Micro-level Contrastive Learning ($L_{MicroCL}$) constitute pivotal elements of graph adaptation, and excluding either component leads to noticeable performance degradation. Likewise, performance exhibits a gradual decline when confidence-aware annotation ($conf$) is excluded, highlighting its necessity. In comparison, our method incorporates all components into a test-time adaptation paradigm and outperforms all alternatives by a significant margin. Notably, T^2ARD surpasses the performance of T^3RD even with only graph adaptation, providing additional evidence of the effectiveness of T^2ARD.

The Alternative Graph Adaptation Strategies. As we have discussed in Section 3.3, there exist numerous self-supervised strategies to perform graph adaptation. In this context, we introduce two additional self-supervised learning strategies for graph adaptation. Although self-supervised learning has been widely studied in the literature (Jin et al., 2022; Chen et al., 2022), most existing methods only consider optimizing the graph structure and node features, which are not aligned with our rumor detection task. Accordingly, we adopt two recent models, Entropy and SLAPS (Fatemi et al., 2021), both leveraging self-supervised learning for graph refinement. The Entropy strategy reduces prediction uncertainty by minimizing the entropy

Target	Twitter-COVID19	Weibo-COVID19
Model	Acc.	Acc.
T^2ARD w/o L_M	0.822	0.802
T^2ARD$_{\text{Llama3.1-8B}}$	0.829	0.814
T^2ARD$_{\text{Qwen2.5-72B}}$	0.832	0.821
T^2ARD$_{\text{GPT-4o}}$	**0.854**	**0.833**

Table 4: Accuracy of pseudo-labels annotated by different LLMs.

Target	Twitter-COVID19	Weibo-COVID19
Model	Acc.	Acc.
T^2ARD w/o L_M	0.822	0.802
T^2ARD$_{\text{base}}$	0.832	0.821
T^2ARD$_{\text{conf}}$	0.843	0.826
T^2ARD$_{\text{conf_gcn}}$	0.821	0.810
T^2ARD$_{\text{conf_reason}}$	**0.854**	**0.833**

Table 5: Accuracy of pseudo-labels annotated by LLMs across different prompts.

of model prediction. Similarly, SLAPS employs a denoising autoencoder loss as a self-supervised objective. Table 3 compares the performance of various graph adaptation strategies. As shown, both Entropy loss and SLAPS Loss underperform relative to our strategy. Notably, Entropy occasionally exhibits inferior performance compared to the **P**re-trained **R**umor **D**etection **M**odel (PMRD), particularly because maximizing the similarity for connected nodes is meaningless for rumor detection. In contrast, our strategy focus on capture the intrinsic traits of social media conversations and exhibits high versatility. More qualitative analyses of hyperparameters are provided in Appendix D.

Pseudo-Label Generation Across Different LLMs. We incorporate LLMs as annotators to assign pseudo-labels for the target graph data. To examine the effect of employing different LLMs for pseudo-labels generation within T^2ARD, we perform a comparative analysis across multiple LLMs, as shown in Table 4. The source dataset is the Twitter dataset. The results show that GPT-4o attains the best performance, and T^2ARD is implemented based on GPT-4o.

Pseudo-Label Generation Across Different Prompts. Table 5 shows the accuracy of pseudo-labels provided by GPT-4o across different prompts on the target datasets. The source dataset is the Twitter dataset. As we can see, "conf_reason" prompt works best performance consistently on

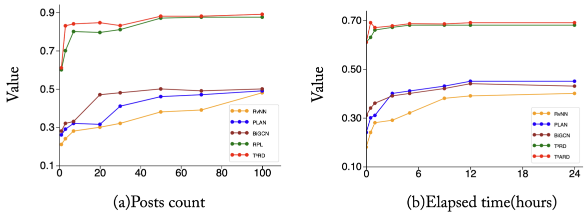


Figure 4: Early detection performance is assessed at various checkpoints based on the count of posts (or elapsed time) on both the Weibo-COVID19 (a) and Twitter-COVID19 (b) datasets.

all target datasets. The "conf" prompt performs better than the "base" prompt, proving the importance of incorporating confidence scores in pseudo-label generation. The "gcn" prompt, which combines the predictions from the pre-trained rumor detection model, fails to consistently yield positive results across all datasets. Accordingly, we adopt the "conf_reason" prompt to achieve more generalizable and superior performance.

4.4 Early Detection

Early rumor alerts are crucial for mitigating the widespread dissemination of rumor content. Detection checkpoints are defined by "delays" measured either as the number of reply posts or as the time elapsed since the initial post. For evaluation, only contents posted at or before each checkpoint is visible to the model. Performance is assessed using the macro-F1 score at each checkpoint. To adhere to each checkpoint, we incrementally scan the test data in chronological order until the target time delay or post count is reached. As illustrated in Figure 4, we compare our method with RvNN, PLAN, BiGCN and T^{3}RD across multiple checkpoints. The proposed T^{2}ARD surpasses other approaches throughout the entire life cycle and achieves a relatively high macro-F1 score at an early stage. Our method requires about 15 posts on Weibo-COVID19 and 3 hours on Twitter-COVID19 to achieve stable performance, while the state-of-the-art method T^{3}RD requires 20 posts to achieve a similar level of performance. These results substantiate our method's strong early detection capability. Additionally, early-stage performance tends to exhibit more or less fluctuation. This is due to the increase in semantic and structural information as statements propagate, resulting in a corresponding increase in noise.

5 Conclusion

In this paper, we propose a novel Test-Time Adaptation framework for Rumor Detection (T^{2}ARD), to address the challenges of cross-domain rumor detection. Our framework leverages both graph adaptation and model adaptation to mitigate distribution shifts between source and target domains. Specifically, we introduce a multi-level self-supervised contrastive learning approach to refine graph structures and node features. Extensive experiments conducted on four widely used cross-domain rumor detection datasets demonstrate that T^{2}ARD achieves state-of-the-art performance. These results highlight the potential of T^{2}ARD as a robust and efficient solution for cross-domain rumor detection tasks.

6 Limitations

Despite the promising results, our proposed T^{2}ARD framework has several limitations. First, the reliance on LLMs for pseudo-label generation introduces computational overhead, which may not be feasible in resource-constrained environments. Moreover, the quality of pseudo-labels depends on the accuracy and confidence estimates of the LLMs, which can vary across different datasets and domains. Second, while our graph adaptation module effectively captures domain-invariant features, it may struggle with highly noisy or incomplete graph structures, which are common in real-world social media data. Lastly, the framework assumes access to a pre-trained rumor detection model, which may not always be available or optimal for certain domains. Future work could explore more efficient and lightweight alternatives for pseudo-label generation, as well as methods to enhance robustness against noisy or sparse graph data.

7 Ethics Statement

This paper adheres to the ACM Code of Ethics and Professional Conduct. The datasets employed are public benchmarks that contain no sensitive private information, and we strictly comply with platform terms and data licenses while avoiding any profiling. We provide proper attribution to prior research, pre-trained model sources, and all toolkits, with complete citations to ensure transparency and reproducibility. The proposed rumor detection approach is intended to enhance the safety and stability of online information ecosystems and public discourse.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62206137, 62206200, 62276257, 62036012, in part by the Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region under Grant NJYT23105, and in part by the National Natural Science Foundation of Inner Mongolia under Grant 2025JQ012.

References

Wenxuan Bao, Zhichen Zeng, Zhining Liu, Hanghang Tong, and Jingrui He. 2024. Adarc: Mitigating graph structure shifts during test-time. *arXiv preprint arXiv:2410.06976*.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 549–556.

Guanzi Chen, Jiying Zhang, Xi Xiao, and Yang Li. 2022. Graphtta: Test time adaptation on graph neural networks. *arXiv preprint arXiv:2208.09126*.

Xuelong Chen and Nan Wang. 2020. Rumor spreading model considering rumor credibility, correlation and crowd classification based on personality. *Scientific reports*, 10(1):5887.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2023. Exploring the potential of large language models (llms) in learning on graph. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Chaoqun Cui and Caiyan Jia. 2024. Propagation tree is not deep: Adaptive graph contrastive learning approach for rumor detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 73–81.

Chaoqun Cui and Caiyan Jia. 2025. Towards real-world rumor detection: Anomaly detection framework with graph supervised contrastive learning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7141–7155.

Yasan Ding, Bin Guo, Yan Liu, Yao Jing, Maolong Yin, Nuo Li, Hao Wang, and Zhiwen Yu. 2025. Evolvedetector: Towards an evolving fake news detector for emerging events with continual knowledge accumulation and transfer. *Information Processing & Management*, 62(1):103878.

Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. 2021. Slaps: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems*, 34:22667–22681.

Shurui Gui, Xiner Li, and Shuiwang Ji. 2024. Active test-time adaptation: Theoretical analyses and an algorithm. *arXiv preprint arXiv:2404.05094*.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. 2023. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*.

hongjian Han and jiaxu Li. 2021. Prediction of information dissemination scale in large-scale online social networks based on machine learning: A comparative analysis. In *Proceedings of the 2021 4th International Conference on Computing and Big Data*, pages 21–26.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. *arXiv preprint arXiv:2305.19523*.

Sergey Ioffe. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. 2022. Empowering graph representation learning with test-time graph transformation. *arXiv preprint arXiv:2210.03561*.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8783–8790.

Jian Liang, Ran He, and Tieniu Tan. 2024. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pages 1–34.

Jian Liang, Dapeng Hu, and Jiashi Feng. 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR.

Hongzhan Lin, Jing Ma, Liangliang Chen, Zhiwei Yang, Mingfei Cheng, and Guang Chen. 2022. Detect rumors in microblog posts for low-resource domains via adversarial contrastive learning. *arXiv preprint arXiv:2204.08143*.

Hongzhan Lin, Jing Ma, Ruichao Yang, Zhiwei Yang, and Mingfei Cheng. 2024. Towards low-resource rumor detection: Unified contrastive transfer with propagation structure. *Neurocomputing*, 578:127438.

Hongzhan Lin, Pengyao Yi, Jing Ma, Haiyun Jiang, Ziyang Luo, Shuming Shi, and Ruifang Liu. 2023. Zero-shot rumor detection with propagation structure via prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5213–5221.

Qiang Liu, Junfei Wu, Shu Wu, and Liang Wang. 2024. Out-of-distribution evidence-aware fake news detection via dual adversarial debiasing. *IEEE Transactions on Knowledge and Data Engineering*.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *The world wide web conference*, pages 3049–3055.

Jing Ma, Jun Li, Wei Gao, Yang Yang, and Kam-Fai Wong. 2021. Improving rumor detection by promoting information campaigns with transformer-based generative adversarial learning. *IEEE Transactions on Knowledge and Data Engineering*.

Haitao Mao, Lun Du, Yujia Zheng, Qiang Fu, Zelin Li, Xu Chen, Shi Han, and Dongmei Zhang. 2024. Source free graph unsupervised domain adaptation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 520–528.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. 2022. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Sikandar Ali Qalati, Mohd Farid Mohd Sharif, Mohd Khairie Ahmad, and Dragana Ostic. 2023. *Proceedings of the 2023 3rd International Conference on Social Development and Media Communication (SDMC 2023)*, volume 807. Springer Nature.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. 2020. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551.

Mengzhu Sun, Xi Zhang, Jiaqi Zheng, and Guixiang Ma. 2022. Ddgcn: Dual dynamic graph convolutional networks for rumor detection on social media. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4611–4619.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR.

Wei Tang, Zuyao Ma, Haifeng Sun, and Jingyu Wang. 2023. Learning sparse alignments via optimal transport for cross-domain fake news detection. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xiang Tao, Mingqing Zhang, Qiang Liu, Shu Wu, and Liang Wang. 2024. Out-of-distribution rumor detection via test-time adaptation. *arXiv preprint arXiv:2403.17735*.

Lin Tian, Xiuzhen Jenny Zhang, and Jey Han Lau. 2022. Duck: Rumour detection on social media by modelling user and comment propagation networks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4939–4949.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341*.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36.

Yiqi Wang, Chaozhuo Li, Wei Jin, Rui Li, Jianan Zhao, Jiliang Tang, and Xing Xie. 2022. Test-time training for graph neural networks. *arXiv preprint arXiv:2210.08813*.

Jiaying Wu and Bryan Hooi. 2023. Decor: Degree-corrected social graph refinement for fake news detection. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2582–2593.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, Yongfeng Zhang, et al. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*, 4(5):7.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. 2017. A convolutional approach for misinformation identification. In *IJCAI*, pages 3901–3907.

Shuo Yu, Yingbo Wang, Ruolin Li, Guchun Liu, Yanming Shen, Shaoxiong Ji, Bowen Li, Fengling Han, Xiuzhen Zhang, and Feng Xia. 2025. Graph2text or graph2token: A perspective of large language models for graph learning. *arXiv preprint arXiv:2501.01124*.

Huaiwen Zhang, Xinxin Liu, Qing Yang, Yang Yang, Fan Qi, Shengsheng Qian, and Changsheng Xu. 2024a. T3rd: Test-time training for rumor detection on social media. In *Proceedings of the ACM on Web Conference 2024*, pages 2407–2416.

Jiaxin Zhang, Yiqi Wang, Xihong Yang, Siwei Wang, Yu Feng, Yu Shi, Ruichao Ren, En Zhu, and Xinwang Liu. 2024b. Test-time training on graphs with large language models (llms). In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2089–2098.

Jiaxin Zhang, Yiqi Wang, Xihong Yang, and En Zhu. 2024c. A fully test-time training framework for semi-supervised node classification on out-of-distribution graphs. *ACM Transactions on Knowledge Discovery from Data*, 18(7):1–19.

Pengfei Zhang, Hongyan Ran, Caiyan Jia, Xuanya Li, and Xueming Han. 2021. A lightweight propagation path aggregating network with neural topic model for rumor detection. *Neurocomputing*, 458:468–477.

Zhen Zhang, Meihan Liu, Anhui Wang, Hongyang Chen, Zhao Li, Jiajun Bu, and Bingsheng He. 2024d. Collaborate to adapt: Source-free graph domain adaptation via bi-directional adaptation. In *Proceedings of the ACM Web Conference 2024*, pages 664–675.

Junyou Zhu, Chao Gao, Ze Yin, Xianghua Li, and Jürgen Kurths. 2024. Propagation structure-aware graph transformer for robust and interpretable fake news detection. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4652–4663.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*, pages 2069–2080.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *Acm Computing Surveys (Csur)*, 51(2):1–36.

A Datasets

We use the TWITTER (Ma et al., 2017) and WEIBO (Ma et al., 2016) datasets as the source data in our cross-domain rumor detection task; In terms of Twitter-COVID19 and Weibo-COVID19 as the target datasets. The statistics of the four real-world cross-domain datasets are shown in Table 6.

B Prompts

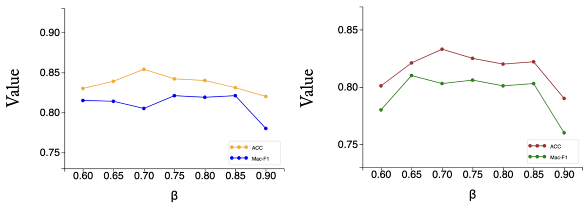
We show the prompts designed for Confidence-aware Annotation (in section 3.4). LLMs were instructed to generate a Python dictionary–like object to simplify the extraction of results from the output text. Guidance for annotation generation was provided via prompt summarized in Table 7. Specifically, we consider four prompt variants: (1) base; (2) conf; (3) conf_gcn; (4) conf_reason. Briefly speaking, "base" assumes no access to ground-truth label information, whereas "conf" augments "base" with a confidence score of LLM-generated pseudo-labels. In addition, "conf_gcn" further incorporates the information from the pre-trained GNN model based on "confidence". The "conf_reason" prompt is a chain-of-thought version of the prompt "conf", which requires the model to first present the reasons for the evaluation and then provide the final decision.

C Implementation Details

We adopt the Bi-directional Graph Convolutional Networks (BiGCN) (Bian et al., 2020) as the pre-trained rumor detection backbone in our framework, fixing the node representation dimension of 128 for all baselines. The optimal learning rate and weight decay are searched in {0.1, $1e^{-2}$, $1e^{-3}$, $1e^{-4}$, $1e^{-5}$}, feature adaptation in {$5e^{-3}$, $1e^{-3}$, $1e^{-4}$, $1e^{-5}$, $1e^{-6}$}, and structure adaptation in {0.5, 0.1, 0.01}. The hyper-parameter α is set as 0.6. All experiments are executed on an NVIDIA RTX

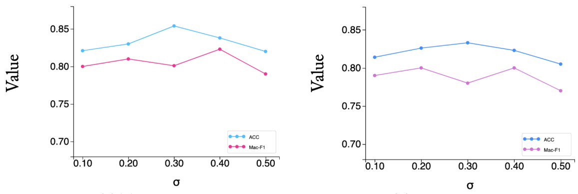
Statistics	Source TWITTER	Target Twitter-COVID19	Source WEIBO	Target Weibo-COVID19
# of events	1154	400	4649	399
# of tree nodes	60409	406185	1956449	26687
# of non-rumors	579	148	2336	146
# of rumors	575	252	2313	253
Avg.# of posts/tree	52	1015	420	67

Table 6: Statistics of the datasets in this paper.



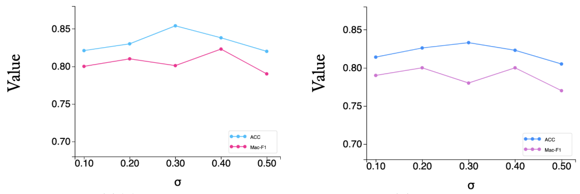
(a) Twitter-COVID19 (b) Weibo-COVID19

Figure 5: Parameter analysis for the β on Weibo-COVID19 and Twitter-COVID19.



(a) Twitter-COVID19 (b) Weibo-COVID19

Figure 7: Parameter analysis for the σ on Weibo-COVID19 and Twitter-COVID19.



(a) Twitter-COVID19 (b) Weibo-COVID19

Figure 6: Parameter analysis for the γ on Weibo-COVID19 and Twitter-COVID19.

3090 GPU, with an average iteration time of approximately 5 hours.

D Qualitative Analysis

D.1 The Effect of Confidence Threshold

For mitigate the biases of LLM-generated pseudo-labels, we request the prediction confidence for the pseudo-labels from LLMs in section 3.4. We conducted experiments to evaluate the accuracy of pseudo-labels under different confidence scores. The confidence threshold is denoted as β. Figure 5 shows that the best accuracy is achieved at $\beta = 0.70$ for all target datasets.

D.2 Impact of Noise Levels in Data Augmentation

We used DropEdge (Rong et al., 2019) and Mask Node Attributes (You et al., 2020) as the data augmentation methods to obtain the augmented view (in section 3.3). Figure 6 and Figure 7 investigate the influence of the dropedge ratios and mask node attributes ratios on performance. We use Twitter as the source dataset. Specifically, we adopt ratios in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ for DropEdge and Mask Node Attributes. The dropedge threshold is denoted as γ. The mask node attributes threshold is denoted as σ. From the Figure 6 and Figure 7, we observe that the optimal results are achieved at $\gamma = 0.2$ and $\sigma = 0.3$ for all target datasets. T^2ARD with either of the two augmentations can greatly improve the performance of generalization.

D.3 Case Study

T^{2}ARD leverages both graph adaptation and model adaptation to mitigate distribution shift. To better understand graph propagation structures and LLM responses, we present an example of a correctly detected rumor along with part of its propagation structure. As shown in Figure 8, posts that challenge a rumor tend to receive supportive replies that affirm the denial, whereas posts endorsing a rumor often trigger refutations. Replies generally address their immediate parent node rather than the root claim. This observation aligns with our motivation to explore rumor propagation structures for representation learning. By applying graph adaptation, MacroCL aligns node representations with global graph representations to capture domain-invariant controversy signals. Since these dissemination patterns generalize across platforms (e.g., Twitter, Weibo), they offer a stable anchor despite surface text variation. MicroCL boosts local robustness by enforcing consistency across lightly perturbed views, reducing sensitivity to cross-domain shifts. During model adaptation, the LLM assigns high-confidence pseudo-labels with reasoning that highlights missing verification, dense refutations, and "fake news" cues. Because the confidence score exceeds the threshold β, the sample is used for lightweight supervised updates to counter drift. Crucially, the agreement between the LLM's semantics-aware reasoning and graph adaptation's structure-aware signals reduces cross-domain error and yields stable rumor predictions.

D.4 Error analysis

To gain deeper insights into model behavior, we analyze a rumor case misclassified by our framework. As shown in Figure 9, the LLM's response reveals an overreliance on surface coherence and repeated retellings, which it mistakenly treats as evidence of truth. Moreover, it insufficiently handles pragmatic cues such as satire, laughter markers, and hyperbole, which often signal non-factual content. The incorrect prediction for this case stems from the unavoidable noise introduced by the LLM when generating pseudo-labels, leading to misclassification in ambiguous situations. Nevertheless, the overall experimental results indicate that LLMs have a significant positive impact on addressing domain shift in rumor detection tasks, with the benefits outweighing the drawbacks.

E Complexity Analysis

We analyze the computational cost of T^{2}ARD by considering the main stages of the pipeline. Time complexity and symbol definition are shown in table 8. During Pre-training stage, the pre-trained model has a per-forward cost of $O(eh)$ for aggregation and $O(nh^2)$ for transformation; over L layers the time complexity is $O(L(eh + nh^2))$. At test-time adaptation phase, graph adaptation constructs V views and optimizes MacroCL/MicroCL losses over node-level embeddings and graph-level embeddings. The forward cost per view remains of the same order as pre-training, i.e., $O\big(VL(eh + nh^2)\big)$; the main additional overhead comes from similarity computations in MicroCL: $O(b^2n^2h)$ with in-batch negatives or $O(bn^2h)$ with within-graph negatives. Model adaptation with LLM supervision introduces limited computational burden: LLM querying is external to GPU compute and contributes an external latency of qT, while fine-tuning a lightweight classifier on the filtered subset costs $O(mhc)$. After adaptation, inference requires $O\big(bL(eh + nh^2)\big)$ time. As shown in table 8, the overall time complexity of our model is $O\big((V + b)L(eh + nh^2) + \kappa + mhc\big)$. Even in the worst-case scenario, the proposed model maintains acceptable time complexity.

Prompt Name	Prompt Content
base	Role: You have been specially designed for the rumor detection task. Event: \n <Event content>\n Task: \n There are following categories: \n <list of categories>\n What's the category of this Event? Output your answer in the form of a list of python dicts like `["answer":<answer_here>]`.
conf	Role: You have been specially designed for the rumor detection task. Event: \n <Event content>\n Task: \n There are following categories: \n <list of categories>\n What's the category of this event? Output your answer together with a confidence ranging from 0 (lowest confidence) to 1 (highest confidence), in the form of a list of python dicts like `[{"answer":<answer_here>, "confidence":<confidence_here>}]`. Confidence must be a float between 0 and 1 (e.g., 0.87) with up to two decimal places.
conf_gcn	Role: You have been specially designed for the rumor detection task. Event: \n <Event content>\n Task: \n There are following categories: \n <list of categories>\n What's the category of this event? Output your answer together with a confidence ranging from 0 (lowest confidence) to 1 (highest confidence), in the form of a list of python dicts like `[{"answer":<answer_here>, "confidence":<confidence_here>}]`. Confidence must be a float between 0 and 1 (e.g., 0.87) with up to two decimal places. The pseudo-labels generated by BiGCN is: BiGCN[<enent_id>]. The confidence of this pseudo-labels is bigcn_conf[<event_id>]. Use this information to help your prediction.
conf_reason	Role: You have been specially designed for the rumor detection task. Event: \n <Event content>\n Task: \n There are following categories: \n <list of categories>\n What's the category of this event? Output your answer together with a confidence ranging from 0 (lowest confidence) to 1 (highest confidence), in the form of a list of python dicts like `[{"answer":<answer_here>, "confidence":<confidence_here>}]`. Confidence must be a float between 0 and 1 (e.g., 0.87) with up to two decimal places. Please provide your reasoning first before making your final decision.

Table 7: The prompts used in T^{2}ARD.

Processing Step	Time Complexity	Symbol Definition
Pre-training Processing	$O\big(L(eh + nh^2)\big)$	n: Total quantity of nodes e: Total quantity of edges h: The node representation dimension L: The number of graph neural network layers
Graph Adaptation Processing	$O\big(VL(eh + nh^2) + \kappa\big)$	V: The number of views b: The batch size $\kappa = b^2n^2h$ (in-batch negatives), or $\kappa = bn^2h$ (within-graph negatives)
Model Adaptation Processing	$O(mhc)$	m: The number of selected samples h: hidden dim c: classes T: The average response time of LLMs q: The number of LLM queries External latency: qT for q LLM queries (not counted in arithmetic)
Inference Processing	$O\big(bL(eh + nh^2)\big)$	b: The batch size
Total	$O\big((V + b)L(eh + nh^2) + \kappa + mhc\big)$	Plus external latency qT; see κ above

Table 8: Time complexity analysis of different processing steps.

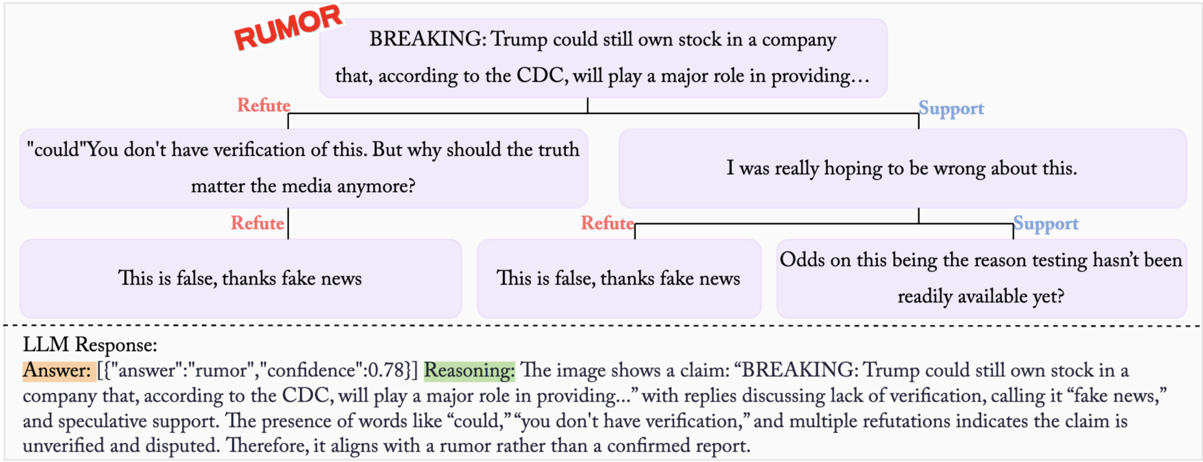


Figure 8: An example case with correct detected rumors of T^{2}ARD.

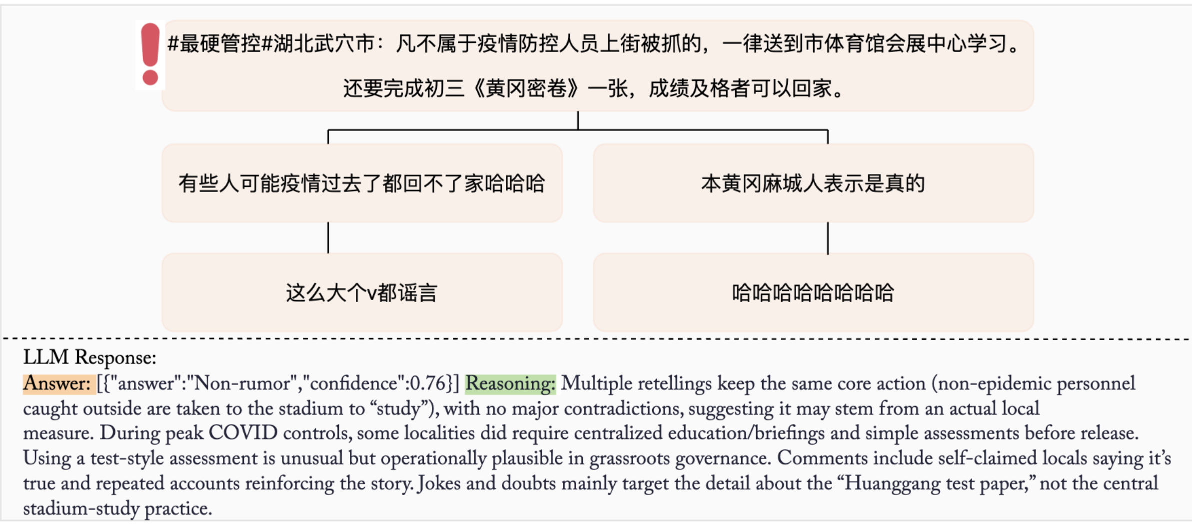


Figure 9: An example case with incorrectly detected rumors of T^{2}ARD.