MONTROSE: LLM-driven Monte Carlo Tree Search Self-Refinement for Cross-Domain Rumor Detection

Shanshan Liu[1,2*], Menglong Lu[1*], Zhen Huang[1§], Zejiang He[1], Liu Liu[3], Zhigang Sun[1], Dongsheng Li[1]

[1]College of Computer Science and Technology, National University of Defense Technology, Changsha, China
[2]The Sixty-Third Research Institute, National University of Defense Technology, Nanjing, China
[3]School of Information Engineering, Suqian University, Suqian, China

{liushanshan17,lumenglong,huangzhen}@nudt.edu.cn

Abstract

With the emergence of new topics on social media as sources of rumor dissemination, addressing the distribution shifts between source and target domains remains a crucial task in cross-domain rumor detection. Existing feature alignment methods, which aim to reduce the discrepancies between domains, are often susceptible to task interference during training. Additionally, data distribution alignment methods, which rely on existing data to synthesize new training samples, inherently introduce noise. To deal with these challenges, a new cross-domain rumor detection method, MONTROSE, is proposed. It combines LLM-driven Monte Carlo tree search (MCTS) data synthesis to generate high-quality synthetic data for the target domain and a domain-sharpness-aware minimization (DSAM) self-refinement approach to train rumor detection models with these synthetic data effectively. Experiments demonstrate the superior performance of MONTROSE in cross-domain rumor detection. The code is available at https://github.com/lisa633/MONTROSE.

1 Introduction

With the advent of machine learning and deep learning techniques, significant progress has been made in rumor detection. These methods effectively mine semantic information from both text content (Ma et al., 2016; Shu et al., 2019; Przybyla, 2020) and propagation structures (Monti et al., 2019; Zhou and Zafarani, 2019; Shu et al., 2020), achieving better rumor detection performance.

However, detecting rumors of emerging topics remains a challenge for existing methods(Yue et al., 2022). Traditional methods detect rumors under in-domain conditions. Nevertheless, in the case of detecting rumors about emerging topics, there is a distribution shift between the emerging topics and historical topics. As shown in Figure 1, there are differences in word usage and writing style between the two domains. If this distributional discrepancy is ignored and the model trained on the source domain is directly applied to the target domain, their performance undergoes a significant diminishment.

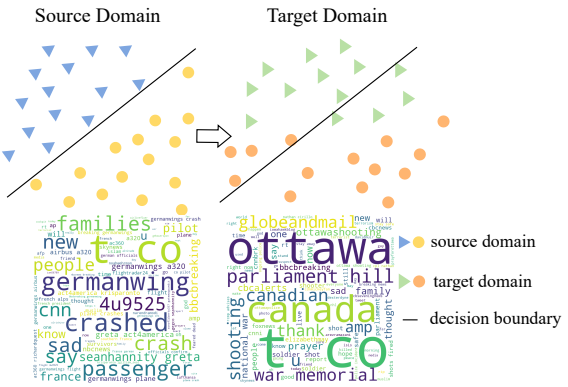

Figure 1: An example of cross-domain rumor detection. The source domain and the target domain exhibit a distribution shift, caused by the variation of word usage and writing style across different topics.

To deal with the distribution shift issue, methods can be divided into two main categories: feature space alignment (Wang et al., 2018; Lin et al., 2022; Shu et al., 2022; Ran and Jia, 2023; Yue et al., 2023) and data distribution alignment (Lu et al., 2023; Shi et al., 2023; Chen et al., 2025; Cui and Jia, 2025). The former methods employ techniques, such as domain adversarial learning (DAL) (Wang et al., 2018) and maximum mean discrepancy (MMD) (Lin et al., 2022), to align the feature space distributions across various topics. However, these methods simultaneously optimize two conflicting objectives, aligning feature space and conducting supervised training, leading to task interference and suboptimal performance (Wang

* contributed equally to this work.
§ corresponding author

Findings of the Association for Computational Linguistics: ACL 2025, pages 21475–21487

et al., 2024). Data distribution alignment aims to create training datasets that closely resemble the target data distribution, enabling model retraining and adaptation through techniques such as pseudo-labeling (Lu et al., 2023) and data selection (Chen et al., 2025). Yet, pseudo-labeling suffers from label noise, and data selection introduces input bias, particularly problematic in complex tasks like rumor detection, where contextual understanding is critical (Zheng et al., 2024; Yan et al., 2024).

In this paper, we propose a new approach for cross-domain rumor detection that combines feature space alignment and data distribution alignment, termed MONTROSE (LL**M**-driven M**ON**te Carlo **T**ree sea**R**ch d**O**main-sharpness-aware minimization **S**elf-r**E**finement). MONTROSE is a two-stage framework that first synthesizes training data with an LLM-Driven Monte Carlo tree search (MCTS) module, and then retrains the model on these synthesized data via a domain-sharpness-aware minimization (DSAM) self-refinement method:

LLM-driven MCTS. Although LLMs excel at generating fluent text, synthesizing rumors remains challenging due to their complex social and psychological underpinnings, which result in intricate contextual and propagation dynamics (Zhao et al., 2024). To address this, we employ MCTS to model the branching nature of rumor dissemination. By identifying key nodes, MCTS guides the LLM generation process, enabling rapid convergence on domain-relevant propagation patterns. This approach ensures that the synthesized data is not only highly readable but also authentically captures the structural characteristics of rumor propagation, thereby achieving the goal of synthesizing domain-specific data.

DSAM Self-Refinement. To refine the model using unlabeled synthetic data, MONTROSE initially assigns pseudo-labels to the data based on the model's high-confidence predictions. However, recent research has shown that while high-confidence samples are generally reliable, they often do not significantly improve model training performance because they tend to reinforce existing model biases (Chen et al., 2022). To mitigate this issue, DSAM self-refinement perturbs the model parameters with the domain alignment loss. By perturbing the parameter space, DSAM prevents overfitting while simultaneously aligning feature space. Meanwhile, since the perturbations are constrained within a bounded range, this mechanism can also effectively mitigate task interference to maintain task-specific performance.

To summarize, our contributions are fourfold:

- We introduce a new framework termed MONTROSE for cross-domain rumor detection. MONTROSE first synthesizes training samples that mimic the target domain's data distribution and then trains the model using a specialized DSAM Self-Refinement algorithm.
- We design an LLM-driven MCTS module to synthesize training data, which recovers the structural characteristics of the rumor propagation through Monte Carlo simulations during the LLM-synthesis process.
- We introduce a DSAM self-refinement training algorithm with a dual mechanism that can perturb the parameter space while aligning the feature space simultaneously.
- Experimental results demonstrate the effectiveness of the proposed MONTROSE framework, validating its superiority in addressing challenges of cross-domain rumor detection.

2 Related Work

2.1 Cross-Domain Rumor Detection

Deep learning-based rumor detection methods effectively mine semantic information from both text content (Ma et al., 2016; Shu et al., 2019; Przybyla, 2020) and propagation structures (Monti et al., 2019; Zhou and Zafarani, 2019; Shu et al., 2020), achieving notable success in detecting rumors under in-domain conditions.

Subsequently, cross-domain rumor detection methods emerged, broadly categorized into two approaches, feature space alignment (Wang et al., 2018; Lin et al., 2022; Shu et al., 2022; Ran and Jia, 2023; Yue et al., 2023) and data distribution alignment (Lu et al., 2023; Shi et al., 2023; Chen et al., 2025; Cui and Jia, 2025). Despite their efforts, existing methods often face challenges such as task interference during training or the introduction of noise during data synthesis, limiting their effectiveness in cross-domain scenarios.

With the rise of LLMs and their outstanding performance, researchers incorporate LLMs into rumor detection (Lai et al., 2024; Ouyang et al., 2024; Hu et al., 2024). However, existing LLM-based methods primarily focus on data augmentation or prompt engineering, and their performance in cross-domain rumor detection remains limited.

2.2 Monte Carlo Tree Search

MCTS is primarily used in game theory and decision-making, which provides a robust framework for the exploration of complex search spaces by simulating and evaluating iteratively (Browne et al., 2012). In recent years, with the rapid development of LLMs, methods that combine MCTS with LLMs have emerged. These methods can maximize the exploration capabilities of large language models, making exploration at different levels possible, and are applied in various areas, e.g., image enhancement (Cotogni and Cusano, 2023), code generation (Brandfonbrener et al., 2024), reasoning (Xie et al., 2024), and synthetic data generation (Locowic et al., 2024).

Inspired by the pivotal nodes in rumor propagation, such as novel or emotionally charged content, we leverage MCTS to identify key nodes and guide domain-oriented data synthesis.

3 Problem Formulation

Cross-domain rumor detection can be denoted as the unsupervised domain adaptation task in text classification. Historic data is the source domain D^S, composed of a feature space $\mathcal{X}^S$, a label space $\mathcal{Y}^S$, and an associated probability distribution $P(X^S, Y^S)$ such that $D^S = \{\mathcal{X}^s, \mathcal{Y}^s, P(X^S, Y^S)\}$. $X^S = \{x_1^S, x_2^S, \ldots, x_n^S\}$ denotes the sample set, where n is the number of samples and x_i^S denotes the input sentence in the text classification task. $Y^S = \{y_1^S, y_2^S, \ldots, y_n^S\}$ is corresponding labels of X^S, where y_i^S is a vector of one hot label of C dimensions, that is, $y_i^S \in \{0,1\}^C$, C is the number of classes. Similarly, samples of emerging topics consist of the target domain $D^T = \{\mathcal{X}^T, \mathcal{Y}^T, Q(X^T, Y^T)\}$. In unsupervised scenarios, the labels of the target domain are not available, and Y^T is unknown. D^S and D^T share the same label space, but as topics change, there exists a domain shift between them. In the case of DA, a hypothesis $h : \mathcal{X}^S \rightarrow \mathcal{Y}^S$ is learned based on D^S. If the same hypothesis h also works for $\mathcal{X}^T \rightarrow \mathcal{Y}^T$ with an acceptable error, the hypothesis h adapts to the target domain and source domain.

4 Methodology

4.1 Overview

The core idea of MONTROSE is to utilize LLMs to synthesize data for new topics and design a training algorithm to effectively use this synthetic data. As shown in Figure 2, MONTROSE consists of three main stages: LLM-driven MCTS data synthesis, pseudo-labeling, and DSAM self-refinement.

In the first stage, we employ LLM-driven MCTS to construct training data that simulates the distribution of the new domain. Since rumor data not only includes text but also encompasses graph structures such as propagation networks, relying solely on LLM for data synthesis fails to capture such propagation characteristics. Therefore, we utilize MCTS to thoroughly explore the possibilities of rumor propagation and guide the generation process with a discriminator to ensure the synthetic data is similar to the target domain.

In the second stage, we generate pseudo-labels for the synthetic data. We predict labels for the newly generated data with the model trained on the source domain. Samples with high confidence scores are selected, and their predicted labels are used as pseudo-labels. However, since these pseudo-labels have high confidence, it is hard to improve training performance.

In the third stage, we conduct training on the pseudo-labeled synthetic data. This method first uses the gradient of a domain alignment loss to perturb the model parameters. On one hand, the perturbed model aligns the feature space. On the other hand, the perturbation process reduces the model's confidence in the pseudo-labeled data, preventing overfitting. Unlike joint loss, the perturbation process allows us to control its magnitude, avoiding interference with the training of the task loss during feature space alignment.

4.2 LLM-Driven Monte Carlo Tree Search

To synthesize high-quality data that maintains the propagation characteristics of rumors, a domain-oriented data synthesis approach based on MCTS and LLM is introduced. A revised MCTS algorithm is used to search available nodes in rumor propagation trees. Each iteration contains four steps:

Selection: Starting from the root node representing the source tweet, we search through the nodes in the rumor propagation tree, which represent replies to the source tweet. The Upper Confidence Bound (UCB) is applied to select nodes:

$$\text{UCB}_i = R_i + \eta\sqrt{\frac{2lnN_p}{N_i}}, \tag{1}$$

where i denotes the i-th node in the tree, and R_i refers to the similarity score to the target domain. η is an exploration parameter that controls the trade-

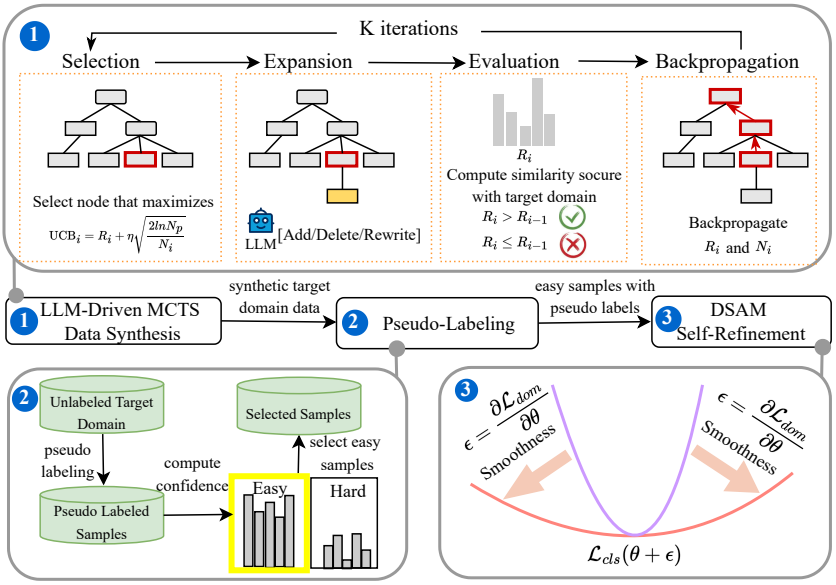

Figure 2: An overview of our proposed MONTROSE, involving LLM-driven MCTS, pseudo-labeling, and DSAM self-refinement. In the beginning, MCTS is applied to select nodes from the rumor tree, while LLM supports add and rewrite actions for data synthesis. Then the synthetic data are pseudo-labeled, and only easy samples with high confidence scores are selected for retraining. The bottom right part shows the perturbation to the task loss with the gradient of the domain alignment loss.

off between exploration and exploitation. N_p represents the total number of times the parent node is modified, while N_i is the number of modification times of the i-th node.

Expansion: After selecting the node, there are three actions to choose: *add, delete, and rewrite*, and the probability of each action being selected is equal. Specifically, *add* indicates adding a reply to the tweet corresponding to the selected node. *Delete* means removing the selected node and the subtree rooted at the selected node. *Rewrite* refers to rewriting the tweet corresponding to the selected node. Adding replies and rewriting tweets are implemented by calling LLMs with prompt learning.

Evaluation: The similarity score R_i is computed in the evaluation stage. After expansion, the modified rumor tree is fed to the domain discriminator trained by samples of the source domain and target domain. Then the softmax function is applied to the output of the discriminator, and its value for the dimension of the target domain is denoted as the similarity score:

$$R_i(k) = \frac{e^{z_k}}{\sum_{j=1}^{C} e^{z_j}}, \tag{2}$$

where C is the number of domains, k represents the target domain, and z_k denotes the output of the domain discriminator in the k-th dimension.

Backpropagation: In the backpropagation process, the similarity score and the number of modification times are propagated from the selected node up to the root of the tree. If the score exceeds the threshold before reaching the maximum number of iterations, the MCTS process is terminated early, and the obtained rumor tree is used as pseudo-data for the target domain.

The example of LLM-driven MCTS is introduced in Appendix A.

4.3 DSAM Self-Refinement

Owing to LLM-driven MCTS data synthesis, we acquire abundant unlabeled pseudo-samples of the target domain. To utilize these samples effectively and realize knowledge transfer from the source domain to the target domain, the DSAM self-refinement approach is introduced, which includes two modules: pseudo-labeling and DSAM perturbation.

Inspired by the self-training, we assign pseudo-labels to samples from data synthesis. Stemming from semi-supervised learning, self-training trains a classifier f_θ on the labeled data from the source domain D^S:$\min_\theta \mathcal{L}_{cls}\left(f(\theta, X^S), Y^S\right)$, where θ denotes parameters of the classifier. Then the trained classifier is used to generate pseudo labels for unlabeled samples in the target domain:

$$\min_{\theta,\hat{\mathbf{Y}}^T} \mathcal{L}_{st}\left(\theta, \hat{\mathbf{Y}}^T\right) = \sum_{x_k, y_k \in D^S} \mathcal{L}_{cls}\left(f\left(\theta, x_k\right), y_k\right) + \sum_{x_i \in D^T} \mathcal{L}_{cls}\left(f\left(\theta, x_i\right), \hat{y}\left(x_i\right)\right), \tag{3}$$

where $\hat{\mathbf{Y}}^T = \{\hat{y_1}^T, \hat{y_2}^T, \ldots, \hat{y_n}^T\}$ represents pseudo labels predicted by classifier f_θ for unlabeled samples in target domain.

To transfer knowledge from the source domain to the target domain, the retraining model is based on domain adversarial learning. The basic components consist of a task-specific model $\mathcal{T}_\Psi$ and a domain discriminator $\mathcal{D}_\Phi$. Specifically, the task-specific model can be divided into two parts, a feature extraction layer g_ψ and a classification layer f_θ. The domain discriminator is applied to learn the discrepancy between D^S and D^T.

The domain adversarial training is to optimize:

$$E(\psi, \theta, \Phi) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{cls}^i(\psi, \theta) - \lambda(\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{dom}^i(\Phi, \theta)), \tag{4}$$

by finding the optimal value of ψ, θ, Φ that

$$\begin{aligned} (\hat{\psi}, \hat{\theta}) &= \arg\min_{\psi,\theta} E(\psi, \theta, \hat{\Phi}), \\ \hat{\Phi} &= \arg\max_{\Phi} E(\hat{\psi}, \hat{\theta}, \Phi). \end{aligned} \tag{5}$$

Although pseudo-labeled high-confidence samples are reliable, they are prone to overfitting and reinforcing the model's existing biases. Inspired by SAM (Foret et al., 2020), we introduce a perturbation to the model parameters with the domain alignment loss, denoted as:

$$\mathcal{L}_{per}(\theta) \triangleq \max_{\|\epsilon\| \le \rho} \mathcal{L}_{cls}(\theta + \epsilon), \tag{6}$$

where $\rho \ge 0$ is a hyperparameter to control the range of perturbation and ϵ is obtained from the domain alignment loss:

$$\epsilon = \frac{\partial \mathcal{L}_{dom}}{\partial \theta}, \tag{7}$$

and domain alignment loss is computed by:

$$\mathcal{L}_{dom} = \frac{1}{\mathcal{B}}\sum_{i=1}^{\mathcal{B}}(\sum_{c=1}^{C} K_c \cdot (-\log(\mathcal{D}(c|x_i;\Phi))) + \sum_{c=1}^{C} \mathcal{D}(c|x_i;\Phi) \cdot \log(\mathcal{D}(c|x_i;\Phi))), \tag{8}$$

where $\mathcal{B}$ represents the training batch, $\mathcal{D}(c|x_i;\Phi)$ is the probability of the instance x_i belongs to the c-th category, and K_c denotes the average probability distribution of the c-th category, computed as:

$$K_c = \frac{1}{\mathcal{B}}\sum_{i=1}^{\mathcal{B}} \mathcal{D}(c|x_i;\Phi). \tag{9}$$

After perturbation, the training loss for the domain adversarial learning is represented as:

$$\mathcal{L}_{DA} = \mathcal{L}_{cls} + \mathcal{L}_{per} - \lambda\mathcal{L}_{dom}. \tag{10}$$

The whole training process is depicted in Algorithm 1, detailed in Appendix B.

5 Experiments

In this section, the dataset, baselines, and implementation details are introduced briefly at first. Following that, experimental results are illustrated.

5.1 Datasets

Our proposed MONTROSE is evaluated on PHEME (Buntain and Golbeck, 2017), Twitter15(Ma et al., 2017) and Twitter16 (Ma et al., 2017). For detailed information about the datasets, please refer to the Appendix C.

5.2 Baselines and Implementation Details

We conduct a comprehensive evaluation of our proposed MONTROSE against various methods.

First, given that MONTROSE is proposed for rumor detection, we compare its performance with several well-designed rumor detection models, including UDGCN (Bian et al., 2020), BiGCN (Bian et al., 2020), and MetaAdapt (Yue et al., 2023). Second, since MONTROSE leverages LLMs to generate readable and contextually synthetic samples, we also compare it with other recent rumor detection methods that utilize LLMs, such as GPT-3.5 (OpenAI, 2022), GPT-4 (Achiam et al., 2023), LLaMA3-7B (Touvron et al., 2023), and ARG (Hu et al., 2024). Third, as rumor detection of emerging topics can be regarded as cross-domain rumor detection, we compare our approach with general domain adaptation methods that share similarities with this scenario, including DANN (Ganin et al., 2016), MME (Saito et al., 2019), BiAT (Jiang et al., 2020), SFT (Chen et al., 2021), WIND (Chen et al., 2021), and DaMSTF (Lu et al., 2023). The details of baseline models can be found in Appendix D. Implementation details (e.g., learning rate, batch size, etc.) are introduced in Appendix E.

5.3 Results

Rumor Detection Performance. To validate the effectiveness of MONTROSE, we compare MONTROSE with existing rumor detection approaches.

Table 1: Results compared with rumor detection methods and LLM-based prompt engineering approaches.

Method	PHEME				Twitter15	Twitter16
	Cha.	Fer.	Ott.	Syd.		
UDGCN	0.641	0.364	0.550	0.506	0.824	0.839
BiGCN	0.672	0.451	0.748	0.657	0.812	0.822
MetaAdapt	0.696	0.589	0.407	0.744	0.357	0.430
Llama3	0.304	0.322	0.443	0.461	0.661	0.673
GPT3.5	0.291	0.364	0.432	0.501	0.630	0.706
GPT4	0.351	0.400	0.623	0.583	0.744	0.785
ARG	0.688	0.590	0.751	0.733	0.835	0.830
MONTROSE	**0.712**	**0.594**	**0.818**	**0.756**	**0.848**	**0.878**

Since MONTROSE incorporates modules based on LLMs, we also compared MONTROSE with LLM-based prompt engineering approaches. Experimental results in Table 1 show that MONTROSE achieved the best performance across all topics, thereby validating its effectiveness in cross-domain rumor detection. Although methods such as UDGCN, BiGCN, and MetaAdapt show good results in traditional rumor detection tasks, their performances deteriorate when directly applied to emerging topics. This indicates that existing methods are not well-suited to handle the unique challenges posed by cross-domain rumor detection. Meanwhile, methods based on LLM prompt engineering generally underperform those based on fine-tuning. This suggests that the performance improvement of our proposed method does not stem from the use of LLMs. In other words, our method's effectiveness can be attributed to MCTS-based domain-oriented data synthesis and DSAM self-refinement.

Table 2: F1 score of cross-domain rumor detection results compared with domain adaptation baselines.

Target	SFT	MME	BiAT	Wind	DANN	DaMSTF	MONTROSE
Cha.	0.586	0.601	0.547	0.552	0.658	0.600	**0.712**
Fer.	0.200	0.081	0.256	0.291	0.542	0.542	**0.594**
Ott.	0.599	0.612	0.614	0.633	0.793	0.694	**0.811**
Syd.	0.424	0.677	0.661	0.628	0.698	0.685	**0.756**

Domain Adaptation Performance. In cross-domain rumor detection, the data distribution of the target domain differs significantly from that of the source domain. This setting is similar to the domain adaptation task. Therefore, we also compare our proposed method with the domain adaptation approaches, and the results are presented in Table 2. It can be found that MONTROSE outperforms all the domain adaptation methods across different target domains. For instance, the F1 score of MONTROSE is higher than that of the best baseline DANN by 5.8% in Syd. This substantial improvement is primarily attributed to two core components of our method: domain-oriented data synthesis and domain-smoothness self-refinement. These components enhance the model's ability to adapt to new domains by enriching the training data and optimizing the learning process.

5.4 Ablation Study

To evaluate the impact of each component of MONTROSE, we conduct an ablation study. In detail, we separately remove the LLM-driven MCTS data synthesis component (***-w/o M***), DSAM perturbation (***-w/o P***), and both components simultaneously (***-w/o M, P***) to monitor changes in performance. The results are detailed in Table 3.

As shown in Table 3, removing either the LLM-driven MCTS data synthesis component or DSAM perturbation results in a degradation of the model's performance. When both components are removed, MONTROSE experiences a substantial performance drop, especially in the Fer. topic, where the F1 score decreases by 13.7%. These results demonstrate the indispensable roles of LLM-driven MCTS data synthesis and DSAM perturbation in improving the model's performance.

Table 3: F1 score of ablation study on cross-domain rumor detection.

Method	PHEME				Twitter15	Twitter16
	Cha.	Fer.	Ott.	Syd.		
MONTROSE	**0.712**	**0.594**	**0.811**	**0.756**	**0.848**	**0.878**
- w/o M	0.676	0.561	0.780	0.728	0.844	0.863
- w/o P	0.678	0.503	0.791	0.727	0.812	0.839
- w/o M, P	0.644	0.457	0.774	0.678	0.807	0.825

5.5 Pareto Analysis

To demonstrate that MONTROSE can alleviate task interference in aligning features, we conducted a

Pareto analysis experiment, comparing the MONTROSE method with the DANN approach, as depicted in Figure 3. The horizontal and vertical axes of the graph represent the task classification error rate and domain classification error rate, respectively. Both MONTROSE and DANN are based on the domain adversarial framework. Within this framework, the goal is to find Pareto optimal solutions whose task classification error rate is as low as possible while domain classification error rate is as high as possible, ideally concentrating results in the 'Golden Area' of the graph. It can be seen from Figure 3, the outcomes for MONTROSE are predominantly located within the 'Golden Area', but the outcomes of DANN are primarily found in the 'Bronze Area'. This comparison underscores the superior performance of MONTROSE in improving the accuracy of task classification and deceiving the domain discriminator.

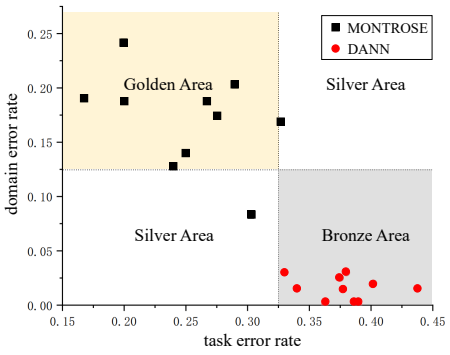

Figure 3: Patero analysis between MONTROSE and DANN. The golden area means a high domain error rate and low task error rate, while the bronze area denotes a low domain error rate and high task error rate. The other area is the silver area.

5.6 Robustness Analysis

Following (Cha et al., 2021), we conduct a local smoothness comparison between MONTROSE and DANN to verify that our proposed MONTROSE can improve robustness. For a given model parameter set θ, we calculate the expected alterations in loss values when transitioning from θ to parameters $\theta^{'}$ situated on a spherical boundary centered at θ with a radius of ϵ, i.e., $\mathcal{F}(\theta) = \mathbb{E}_{\|\theta'\|=\|\theta\|+\epsilon}\left[\mathcal{E}\left(\theta'\right) - \mathcal{E}(\theta)\right]$. In practice, the value of $\mathcal{F}(\theta)$ is estimated using a Monte Carlo sampling approach, with a sample size of 100.

In Figure 4, we compare local smoothness via loss gap $\mathcal{F}(\theta)$ between MONTROSE and DANN, by varying radius ϵ. It can be found that with the increase of ϵ, the curve representing the $\mathcal{F}(\theta)$ of MONTROSE exhibits a more gradual growth trend compared to DANN. This indicates that MONTROSE can find flatter minima, suggesting a potentially more robust performance in the face of increased perturbation.

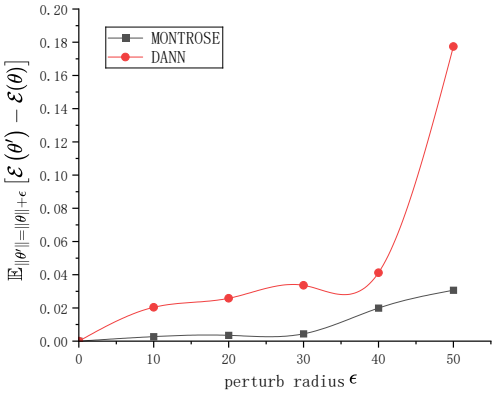

Figure 4: Local smoothness comparison between MONTROSE and DANN.

5.7 DSAM Perturbation Effect

In this section, we conduct a visualization experiment to analyze the effect of DSAM perturbation. This perturbation is based on the domain alignment loss, which is used to perturb the task loss to ensure that the loss landscapes for both task loss and domain loss become smoother. Current research finds that the smoother the loss landscape, the more robust the model (Foret et al., 2020). As presented in Figure 5, after introducing DSAM perturbation, both the task loss and domain loss landscapes become smoother. This visual evidence supports our hypothesis that DSAM perturbation effectively smoothens the loss landscapes. By reducing the sharpness of the loss surfaces, the model is less likely to get stuck in suboptimal local minima. Consequently, this leads to improved performance on both the classification task and domain discrimination objectives, validating the effectiveness of the DSAM perturbation module in enhancing the model's overall generalization.

5.8 T-SNE Visualization

In the cross-domain rumor detection task, there is an inevitable distribution shift between the source domain and the target domain. It is expected that the trained model can correctly distinguish whether samples belong to the source or target domain. To

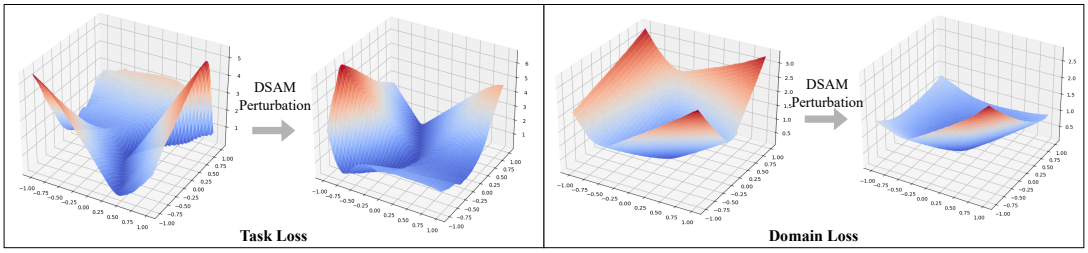

Figure 5: Comparison of the smoothness of task loss and domain loss landscape before and after DSAM perturbation.

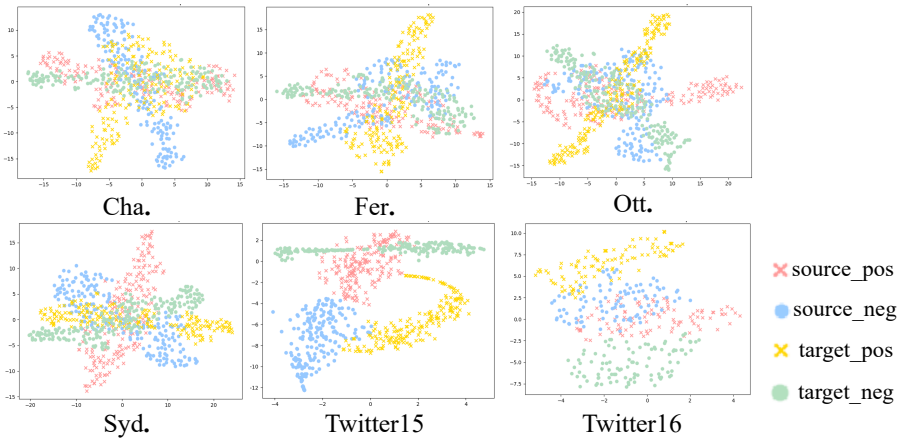

Figure 6: T-SNE visualization of MONTROSE on Pheme, Twitter15, and Twitter16.

analyze this, we performed t-SNE visualizations of the feature representations of the trained MONTROSE when transferred among the PHEME, Twitter15, and Twitter16 datasets. It can be seen from Figure 6 that our proposed MONTROSE can better differentiate samples as to whether they come from the source domain or the target domain. At the same time, it can also accurately distinguish between positive and negative examples within the domain, facilitating better classification tasks.

6 Conclusion

In this paper, we introduced MONTROSE to address the challenges of cross-domain rumor detection. LLM-driven MCTS data synthesis is integrated to generate high-quality synthetic data tailored to the target domain's characteristics. A DSAM self-refinement method is further utilized to perturb model parameters with domain classification gradients, aligning the feature space and enhancing the training contribution of high-confidence samples. Experimental results demonstrate that MONTROSE outperforms existing methods in detecting rumors in emerging topics, making it a robust solution for real-time and cross-domain rumor detection.

Limitations

While MONTROSE presents an effective approach to cross-domain rumor detection, it also has some limitations. First, we assume that the target domain has some underlying similarity to the source domain, which may not always be the case. In scenarios where the domain shift is too drastic, the synthetic data generated by LLM-driven MCTS may not be sufficient to bridge the gap, leading to suboptimal detection performance. Second, MCTS, while powerful, can be resource-intensive and time-consuming, particularly when dealing with large-scale datasets or complex propagation structures. This could hinder the scalability of MONTROSE in real-time or large-scale applications.

Acknowledgments

This work is sponsored in part by the National Natural Science Foundation of China under grant No. 62406332, 62025208, and 62421002.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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.

David Brandfonbrener, Sibi Raja, Tarun Prasad, Chloe Loughridge, Jianang Yang, Simon Henniger, William E Byrd, Robert Zinkov, and Nada Amin. 2024. Verified multi-step synthesis using large language models and monte carlo tree search. *arXiv preprint arXiv:2402.08147*.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Cody Buntain and Jennifer Golbeck. 2017. Automatically identifying fake news in popular twitter threads. In *2017 IEEE international conference on smart cloud (smartCloud)*, pages 208–215. IEEE.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. 2021. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418.

Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. 2022. Debiased self-training for semi-supervised learning. *Advances in Neural Information Processing Systems*, 35:32424–32437.

Songlin Chen, Xiaoliang Chen, Duoqian Miao, Hongyun Zhang, Xiaolin Qin, and Peng Lu. 2025. Ada-uda: A transferable transformer framework for rumor detection using adversarial domain alignment within unsupervised domain adaptation. *Expert Systems with Applications*, 261:125487.

Xiang Chen, Yue Cao, and Xiaojun Wan. 2021. Wind: Weighting instances differentially for model-agnostic domain adaptation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2366–2376.

Marco Cotogni and Claudio Cusano. 2023. Treenhance: A tree search method for low-light image enhancement. *Pattern Recognition*, 136:109249.

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.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35.

Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22105–22113.

Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. 2020. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pages 934–940.

Jianqiao Lai, Xinran Yang, Wenyue Luo, Linjiang Zhou, Langchen Li, Yongqi Wang, and Xiaochuan Shi. 2024. Rumorllm: A rumor large language model-based fake-news-detection data-augmentation approach. *Applied Sciences*, 14(8):3532.

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*.

Leonardo Locowic, Alessandro Monteverdi, and Eleazar Mendoza. 2024. Synthetic data generation from real data sources using monte carlo tree search and large language models. *Authorea Preprints*.

Menglong Lu, Zhen Huang, Yunxiang Zhao, Zhiliang Tian, Yang Liu, and Dongsheng Li. 2023. Damstf: Domain adversarial learning enhanced meta self-training for domain adaptation. *arXiv preprint arXiv:2308.02753*.

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. In *IJCAI*, pages 3818–3824.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Association for Computational Linguistics*, pages 708–717.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

TB OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. In *OpenAI*.

Yi Ouyang, Peng Wu, and Li Pan. 2024. Cool: Comprehensive knowledge enhanced prompt learning for domain adaptive few-shot fake news detection. *arXiv preprint arXiv:2406.10870*.

Piotr Przybyla. 2020. Capturing the style of fake news. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 490–497.

Hongyan Ran and Caiyan Jia. 2023. Unsupervised cross-domain rumor detection with contrastive learning and cross-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13510–13518.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. 2019. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058.

Yu Shi, Xi Zhang, Yuming Shang, and Ning Yu. 2023. Don't be misled by emotion! disentangle emotions and semantics for cross-language and cross-domain rumor detection. *IEEE Transactions on Big Data*.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 395–405.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. 2020. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 626–637.

Kai Shu, Ahmadreza Mosallanezhad, and Huan Liu. 2022. Cross-domain fake news detection on social media: A context-aware adversarial approach. In *Frontiers in fake media generation and detection*, pages 215–232. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jiandong Wang, Hongguang Zhang, Chun Liu, and Xiongjun Yang. 2024. Fake news detection via multiscale semantic alignment and cross-modal attention. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2406–2410.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*.

Yeqing Yan, Peng Zheng, and Yongjun Wang. 2024. Enhancing large language model capabilities for rumor detection with knowledge-powered prompting. *Engineering Applications of Artificial Intelligence*, 133:108259.

Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. 2022. Contrastive domain adaptation for early misinformation detection: A case study on covid-19. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2423–2433.

Zhenrui Yue, Huimin Zeng, Yang Zhang, Lanyu Shang, and Dong Wang. 2023. Metaadapt: Domain adaptive few-shot misinformation detection via meta learning. *arXiv preprint arXiv:2305.12692*.

Shouhao Zhao, Shujuan Ji, Jiandong Lv, and Xianwen Fang. 2024. Propagation tree says: dynamic evolution characteristics learning approach for rumor detection. *International Journal of Machine Learning and Cybernetics*, pages 1–17.

Peng Zheng, Yong Dou, and Yeqing Yan. 2024. Sensing the diversity of rumors: Rumor detection with hierarchical prototype contrastive learning. *Information Processing & Management*, 61(6):103832.

Xinyi Zhou and Reza Zafarani. 2019. Network-based fake news detection: A pattern-driven approach. *ACM SIGKDD explorations newsletter*, 21(2):48–60.

Table 4: Mathematic Symbol List

D^S	source domain
D^T	target domain
$\mathcal{X}^S/\mathcal{X}^T$	feature space
$\mathcal{Y}^S/\mathcal{Y}^T$	label space
$P(X,Y)$	probability distribution
x_i	input sentence
y_i	vector of one-hot label
$\mathcal{T}_\Psi$	task-specific model
$\mathcal{D}_\Phi$	domain discriminator
g_ψ	feature extraction layer
f_θ	classification layer
$\mathcal{L}_{cls}$	rumor detection loss
$\mathcal{L}_{dom}$	domain discrimination loss
$\mathcal{L}_{per}$	perturbation loss in DSAM
ϵ	perturbation obtained form domain alignment
ρ	hyperparameter to control the range of perturbation
λ	hyperparameter to control the proportion of domain discrimination loss

A Example of Domian-Oriented Data Synthesis

For a better understanding of the data synthesis process, we present an example here. The samples used for generation are from the source domain of PHEME and Twitter16, and the target domain is Twitter15. For one sample in the source domain, we first construct a rumor propagation tree based on the retweeting relationships between tweets. For example, the sample has one source tweet, which is retweeted three times. In this case, node 0 represents the original tweet as the root node, and nodes 1,2,3, which represent the retweets, are the child nodes of node 0. Based on this tree structure, we start the MCTS from the root node.

At first, we use the trained domain discriminator to classify the current rumor propagation tree and take the value of the softmax function in the target domain dimension as the initial similarity score. In the selection phase, we traverse the child nodes of the root node and select the child node with the highest UCB score computed by Equation 1. Since it is the first iteration, the UCB values of nodes 1, 2, and 3 are all positive infinity. Therefore, following the principle of depth-first search, we first select node 1. In the expansion phase, we randomly choose an action from: *add, delete, and rewrite*. For *add* action, we utilize the tweet represented by node 1 as [prompt sentence] and randomly select one tweet from the target domain as [target sentence]. We utilize LLM to synthesize a retweet for the selected node with the prompt:

> **Context**: You are a Twitter user. You can generate a Twitter-form reply to make it look like replies in the target domain.
> **Prompt**: Here is an example in the target domain: [target sentence]. The given Twitter is: [prompt sentence]. Please generate a target-domain-form reply to the given Twitter.

For the *delete* action, we remove the selected node directly. It is worth noting that if the node to be deleted has descendant nodes, we delete the node and all its descendant nodes. As for the *modify* action, [prompt sentence] and [target sentence] are the same as the *add* action. LLM is utilized to rephrase the tweet represented by the selected node with the prompt:

> **Context**: You are a Twitter user. You can rephrase a Twitter-form reply to make it like replies in the target domain.
> **Prompt**: Here is an example in the target domain: [target sentence]. The given Twitter is:[prompt sentence]. Please rephrase the given Twitter to make it a reply in the target domain.

Assuming that we choose *add*, we add node 4 with the synthetic tweet as the child node for selected node 1. In the evaluation phase, we use the trained domain discriminator and the updated tree from the expansion step to calculate the similarity score R_i, which is the maximum value of the softmax function in the target domain dimension. If the similarity score is higher than the initial similarity score, we accept this modification, increment the modification counts of node 1 and its parent node by 1, respectively, and update the similarity score of node 1. Otherwise, we reject this modification and proceed directly to the next iteration. We repeat the above steps in a loop until the similarity score exceeds the threshold or the number of iterations reaches the preset maximum value.

B Algorithm of MONTROSE

The detailed pseudo-code of MONTROSE is shown in Algorithm 1.

C Dataset

The PHEME dataset (Buntain and Golbeck, 2017) is a collection of Twitter conversation threads about rumors. It contains 330 labeled source tweets across five topics: Charlie Hebdo (Cha.), Ferguson (Fer.), Germanwings Crash (Ger.), Ottawa Shooting (Ott.), and Sydney Siege (Syd.), with each source tweet having a tree of replies and further interactions, resulting in a total of 4,512 additional descendant tweets. PHEME is annotated by journalists to label the truthfulness of the rumors as true or false. The Twitter15 and Twitter16 datasets (Ma et al., 2017) are collections of microblog posts and their propagation structures used for rumor detection research, both annotated with labels for non-rumors, false rumors, true rumors, and unverified rumors. To align with the PHEME, we label non-rumors in the Twitter15 and Twitter16 datasets as false, and the other three categories (false rumors, true rumors, and unverified rumors) as true.

D Baselines

D.1 Rumor Detection Baselines

- UDGCN (Bian et al., 2020): UDGCN is a graph-based model for rumor detection that uses an undirected graph structure to capture the relationships among posts in a rumor propagation tree, aggregating information from neighboring nodes to learn high-level representations for identifying rumors.

- BiGCN (Bian et al., 2020): Compared with UDGCN, BiGCN can capture both the top-down propagation patterns and bottom-up dispersion structures of rumors by integrating two GCNs to learn comprehensive high-level representations for rumor identification.

- MetaAdapt (Yue et al., 2023): MetaAdapt is a meta-learning-based approach for domain adaptive few-shot misinformation detection, which leverages limited target examples to guide the transfer of knowledge from source to target domains by adaptively learning from source tasks and optimizing model performance in the target domain.

Algorithm 1 MONTROSE

Require: labeled source dataset D^S, unlabeled target dataset D^T, pre-trained LLM

Pretrain θ on D^S
$D^T_{pse} \leftarrow$ MCTSGENERATION(D^S,D^T,θ)
$D^T_{uni} = D^T \cup D^T_{pse}$
while the termination criteria is not met **do**
  Compute pseudo label $\hat{\mathbf{Y}}^T$ on D^T_{uni}
  $H = -\hat{\mathbf{Y}}^T * \log(\hat{\mathbf{Y}}^T)$
  Sort the samples whose H is higher than the threshold to construct D^T_{tra}
  DOMAINPERTURATION(D^S,D^T_{tra},ψ, θ, Φ)
function MCTSGENERATION(D^S,D^T,θ)
  Initialize tree $\mathcal{T}$ for D^S_k
  Set $i \leftarrow 0$,$j \leftarrow 0$
  while $j < M$ **do**
    Select node n_i from $\mathcal{T}$ with the highest UCB_i
    $\mathcal{T}' \leftarrow$ expand node n_i by selecting action randomly from the action list [add, delete, rewrite] powered by LLM
    Compute similarity score R_i with θ
    if $R_i > threshold$
      return $\mathcal{T}'$
    else if
      if $R_i > R_{i-1}$
        $N_i += 1$
        $N_p += 1$
      end if
    end if
end function
function PERTURATION(D^S,D^T_{tra},ψ, θ, Φ)
  for training batch $\mathcal{B}$ in $D^S \cup D^T_{tra}$ **do**
    for $t = 1 \rightarrow \mathcal{T}_{dom}$ **do**
      $\Phi = \Phi - \eta_1 \nabla_\Phi \mathcal{L}_{DA}(\psi, \theta, \Phi, \mathcal{B})$
    end for
    for $t = 1 \rightarrow \mathcal{T}_{per}$ **do**
      $\theta = \theta + \nabla_\theta \mathcal{L}_{dom}(\theta, \mathcal{B})$
    end for
    for $t = 1 \rightarrow \mathcal{T}_{cls}$ **do**
      $\psi = \psi + \eta_2 \nabla_\psi \mathcal{L}_{DA}(\psi, \theta, \Phi, \mathcal{B})$
    end for
end function

D.2 LLM-based Baselines

We selected three representative LLMs as baselines for rumor detection: GPT-3.5 (OpenAI, 2022) (gpt-3.5-turbo), GPT-4 (Achiam et al., 2023) (gpt-4-1106-preview), and LLaMA-3-7B (Touvron et al., 2023) (Llama-3-8B-Instruct). For GPT-3.5 and GPT-4, we utilized their APIs from OpenAI[1]. For LLaMA-3-8B, we used weights from ModelScope[2]. ARG (Hu et al., 2024) is proposed to leverage the multi-perspective rationales generated by LLMs to enhance the performance of small language models by selectively acquiring useful insights and improving their ability to make accurate judgments.

D.3 Domain Adaptation Baselines

- SFT (Chen et al., 2021): SFT refers to supervised fine-tuning, a training approach where a pre-trained model is further adjusted on a specific target domain using labeled data from that domain, aiming to improve its performance on the target task.

- MME (Saito et al., 2019): MME is an adversarial learning approach for semi-supervised domain adaptation, which optimizes an adaptive few-shot model by alternately maximizing the conditional entropy of unlabeled target data concerning the classifier and minimizing it for the feature encoder, thereby learning discriminative and domain-invariant features.

- BiAT (Jiang et al., 2020): BiAT generates adversarial examples bidirectionally between source and target domains using gradients to guide the perturbations, thereby filling the domain gap and improving model performance.

- WIND (Chen et al., 2021): WIND is a model-agnostic instance weighting algorithm for domain adaptation, which automatically learns optimal instance weights through a bi-level optimization framework inspired by meta-learning, thereby improving model generalization on target domains.

- DANN (Ganin et al., 2016): DANN achieves domain adaptation by learning domain-invariant features through an adversarial training process that aligns feature distributions across different domains.

- DaMSTF (Lu et al., 2023): As a self-training framework for domain adaptation, DaMSTF integrates domain adversarial learning and meta-learning to reduce label noise, preserve hard examples, and improve performance.

E Implementation Details

MONTROSE employs BERT+GCN as the base rumor detection architecture, which is very common in rumor detection methods. For the LLM component, we utilize Qwen-Turbo by calling the API provided by Alibaba. We implement MONTROSE and other baselines applying PyTorch with CUDA 10.0 on Ubuntu 18.04.5 LTS servers with NVIDIA A100 GPU. For optimization, Adam optimizers are utilized across all datasets. We trained the model with a batch size of 32 and initialized the learning rate to $5e-5$. ρ in Equation (6) is set to $5e-6$ and the learning rate for DSAM perturbation is $5e-6$.

We constructed a cross-domain scenario based on existing rumor detection datasets. Specifically, we select the data of one topic as the target domain, while the data of other topics are used as the source domain. As for evaluation metrics, we utilize F1 score for the classification of the 'rumor' category. The experiments on PHEME are conducted on "Cha.", "Fer.", "Ott.", and "Syd."[3]

[1] https://openai.com/

[2] https://modelscope.cn/models/llm-research/meta-llama-3.8b-insturct

[3] The labeled data in the "Ger." topic is too scarce to obtain reliable results