Contrastive Learning with Edge-Wise Augmentation for Rumor Detection

Liu, Nan; Zhang, Fengli; Gao, Qiang; Chen, Xueqin

DOI
10.1155/2024/3858526

Publication date
2024

Document Version
Final published version

Published in
International Journal of Intelligent Systems

Wiley
International Journal of Intelligent Systems
Volume 2024, Article ID 3858526, 16 pages
https://doi.org/10.1155/2024/3858526

WILEY

Research Article

Contrastive Learning with Edge-Wise Augmentation for Rumor Detection

Nan Liu,[1] **Fengli Zhang**,[1] **Qiang Gao**,[2] **and Xueqin Chen**[3]

[1]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China*
[2]*School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, China*
[3]*Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, Netherlands*

Correspondence should be addressed to Xueqin Chen; x.chen-10@tudelft.nl

Received 28 December 2023; Revised 2 July 2024; Accepted 23 July 2024

Academic Editor: Ersin Elbasi

Exploring and modeling the spreading process of rumors have shown great potential in improving rumor detection performance. However, existing propagation-based rumor detection models often overlook the uncertainty of the underlying propagation structure and typically require a large amount of labeled data for training. To address these challenges, we propose a novel rumor detection framework, namely, the Uncertainty-Inference Contrastive Learning (UICL) model. Specifically, UICL innovatively incorporates an edge-wise augmentation strategy into the general contrastive learning framework, including an edge-inference augmentation component and an EdgeDrop augmentation component, which primarily aim to capture the edge uncertainty of the propagation structure and alleviate the sparsity problem of the original dataset. A new negative sampling strategy is also introduced to enhance contrastive learning on rumor propagation graphs. Furthermore, we use labeled data to fine-tune the detection module. Our experiments, conducted on three real-world datasets, demonstrate that UICL can not only significantly improve detection accuracy but also reduce the dependency on labeled data compared to state-of-the-art baselines.

1. Introduction

Over the past decade, the rise of online social networks (OSNs) like Twitter, Weibo, Facebook, and Instagram has revolutionized the information-sharing landscape, bringing unparalleled convenience to our daily lives. However, the proliferation of baseless rumors on OSNs has made it increasingly challenging to distinguish fact from fiction, turning these platforms into potential weapons that can harm society [1, 2]. For instance, during the ongoing COVID-19 pandemic, false assertions have circulated, claiming that fish tank cleaning products can cure the virus or that 5G networks are responsible for its spread. In light of this, it has become imperative to promptly and accurately detect rumors, as it represents a crucial social responsibility to mitigate the detrimental impact of such misinformation on individuals and society as a whole.

Motivated by the achievements of deep learning in computer vision (CV) and natural language processing (NLP), researchers have shifted their focus towards developing deep learning-based models for rumor detection [3]. Unlike conventional approaches that heavily rely on manual feature engineering [4, 5], deep learning-based methods offer a promising alternative. While most existing models in this domain are content-based [6, 7], the resemblance of rumor content to real news poses a significant challenge for detection solely based on content features. To tackle this issue, recent studies have emphasized modeling the underlying propagation structure of rumors. For instance, Vosoughi et al. [8] revealed that rumors spread faster, wider, and deeper than the truth, providing a theoretical basis for leveraging propagation structure in developing effective rumor detection models. This line of research aims to enhance the detection accuracy by incorporating the dynamics of rumor spread into the learning process.

Propagation-based models aim to detect rumors by extracting diffusion patterns from the propagation threads of rumors. Ma et al. [9] employed a tree-structured RNN to capture the underlying rumor representation from propagation structures and reply text for effective rumor detection. Subsequent works, including BiGCN [10], MMRD [1], AGWu-RF [11], and CNFRD [12], have further enhanced performance by incorporating graph neural networks (GNNs) into the detection models. GNNs excel at extracting global structural relationships in rumor dispersion, making them a powerful tool in improving detection accuracy. By leveraging the capabilities of GNNs, these models effectively capture the complex interdependencies within the propagation network, thereby enhancing the ability to identify and classify rumors.

While propagation-based models have shown improvements over content-based approaches, they still have significant limitations. One limitation is that the adjacency matrix they construct only captures the presence or absence of edges, ignoring the changes in edge strength. This oversight fails to account for the diversity and variability of user relationships in social networks, resulting in limited consideration of the uncertainty present in the constructed propagation graphs. In real social networks, the relationships among rumor spreaders are variable, leading to uncertainty in the propagation process. Additionally, training propagation-based models often requires a substantial amount of labeled data, which is obtained through manual efforts and is time-consuming. Hence, overcoming these limitations is crucial for developing more robust and accurate rumor detection models.

To address the aforementioned limitations, we present the Uncertainty-Inference Contrastive Learning (UICL) model. Our model builds upon the approach proposed by Li et al. [13], emphasizing its enhanced capability to capture the inherent uncertainty and diversity within rumor propagation structures, as well as facilitating meaningful comparisons with nonrumors. UICL introduces an innovative edge-wise augmentation module that effectively captures uncertainty in constructed propagation graphs through the utilization of prior probability and adaptive edge weight adjustment. This module seamlessly integrates into a comprehensive graph contrastive learning framework, enabling the acquisition of accurate rumor representations. Moreover, UICL undergoes a fine-tuning process using a minimal amount of labeled data to optimize model training. By incorporating uncertainty-aware augmentation and contrastive learning, UICL offers a promising solution to overcome limitations and enhance the performance of rumor detection models. In sum, the main contributions of our work are

(i) We propose the UICL framework, which can capture the uncertainty of rumor propagation structure and alleviate the dependency on labeled data at the same time.

(ii) We design a novel edge-wise augmentation strategy to replace the conventional enhancement strategy in graph contrastive learning, ensuring the model's ability to capture the edge uncertainty of the propagation structure. In addition, we employ a new negative sampling strategy to obtain discriminative negative samples.

(iii) We conducted extensive experiments on three real-world benchmark datasets, demonstrating that UICL can outperform state-of-the-art baselines and achieve significant improvements in detection accuracy.

The paper is organized as follows: in Section 2, we provide a review of the related work in the fields of rumor detection and graph contrastive learning. Section 3 formalizes the problem of rumor detection and introduces relevant definitions. In Section 4, we delve into the details of our proposed UICL model. The results of our experimental evaluations, quantifying the advantages of our approach, are presented in Section 5. Finally, in Section 6, we draw conclusions based on our findings and discuss potential future directions in the field.

2. Related Work

In this section, we provide a brief review of the related works for rumor detection and graph contrastive learning.

2.1. Rumor Detection. In recent years, deep learning methods have made significant strides in the field of rumor detection by automatically learning and extracting valuable semantic information, surpassing traditional methods [5, 14] that heavily rely on hand-designed features and gradually becoming state-of-the-art methods. These deep-learning-based rumor detection models commence by utilizing Recurrent Neural Networks (RNNs) [15] and Convolutional Neural Networks (CNNs) [16] to extract rich contextual and semantic features from input microblog posts. Subsequently, with the rapid development in the field of deep representation learning, researchers are recognizing the limitations of textual features and have begun exploring the potential of diverse feature perspectives for rumor detection. Simultaneously, they are dedicated to developing bespoke methods using advanced technologies. For instance, Ma et al. [9] established a connection between semantic content and propagation cues, utilizing Recursive Neural Networks (RvNN) for rumor detection. Similarly, Liu et al. [17] combined recurrent and convolutional networks to capture both global and local variations in user characteristics throughout the propagation path, significantly enhancing detection accuracy. Besides, with the merits of the attention mechanism [18] in distinguishing feature importance, Khoo et al. [19] employed attention mechanisms at both the post and token levels to predict the veracity and types of rumors in microblogs, simultaneously offering comprehensive explanations of the predictions. Furthermore, an increasing number of researchers are beginning to explore the fusion of diverse semantics from images and text to enhance rumor detection performance. For example, Zou et al. [20] proposed the CMAC fusion strategy that employs adversarial and contrastive paradigms to align distributions of text and image features, generating modal-invariant multimodal fusion representation. Sun et al. [21] introduced the KDCN model, consisting of two subnetworks, which is utilized to detect cross-modal and content-knowledge inconsistencies, thus supporting robust multimodal learning even with missing visual data. Zheng et al. [22] introduced MFAN for social media rumor detection, integrating textual, visual, and social graph features through multimodal fusion and adversarial training. Some researchers also underscore the importance of user feature information, gaining valuable insights from a user's perspective by incorporating user-related features along with other data sources [23].

The discovery that the diffusion processes of fake and real news vary [8] has led to a series of propagation-based works. Since the emergence of graph neural networks (GNNs) has revolutionized the extraction of features from unstructured data, these works often integrate GNNs into their models. Bian et al. [10] first incorporated top-down and bottom-up GNNs to gain a holistic view of rumor spread. Wei et al. [24] proposed a Fuzzy Graph Convolutional Network (FGCN) model that enhances the representation of complex interactions in an information cascade by constructing a heterogeneous graph structure and an edge fuzzification module. Chen et al. [1] build on the concept of GNN-based models and further incorporate RNNs into their model architecture to simultaneously extract both macroscopic and microscopic diffusion features. Distinct from prior studies, Lin et al. [25] proposed a claim-guided graph attention network that extracts multilevel rumor-indicative features from an undirected conversation-based graph.

Besides that, some researchers have combined techniques such as transfer learning, and few-shot learning, enabling the performance of rumor detection models to be maintained with limited knowledge. Lin et al. [26] proposed a zero-sample rumor detection framework that uses a hierarchical cue coding mechanism and domain-invariant propagation features to detect rumors across various languages and domains. Ran et al. [27] proposed an unsupervised cross-domain rumor detection method that enhances performance by generating pseudolabels through clustering and learning domain-invariant features with a cross-attention mechanism. Additionally, some works seamlessly integrate the self-supervised concept into their models to reduce dependency on labeled data. Qian et al. [28] proposed the GACL model, combining graph contrastive learning and adversarial learning, effectively capturing event-specific features and reducing noise impact. Li et al. [29] introduced noise and utilized contrastive learning with asymmetric structure construction for rumor detection, enhancing robustness and accuracy. The latest works also focus on addressing the low-resource problem in rumor detection. For instance, Lin et al. [30] proposed an adversarial contrastive learning-based rumor detection method that adapts to diverse domains and languages, enhancing robustness through language alignment and supervised contrastive training. Furthermore, they enhanced model performance by introducing undirected topology to represent propagation and employing multiscale GCNs for representing propagation structures [31].

These studies showcase diverse and innovative approaches, leveraging network analysis, attention mechanisms, multitask learning, graph-based techniques, etc., to gain deeper insights and improve detection accuracy in the challenging field of rumor detection.

2.2. Graph Contrastive Learning. Graph Contrastive Learning (GCL) is a promising technique that combines graph-based structures and contrastive learning algorithms to improve rumor detection, and its key idea is to enhance the similarity between similar graph elements by mapping them into similar embedding spaces. By maximizing the similarity between positive samples while minimizing the similarity between negative samples, GCL can learn more discriminative graph representations. Common techniques in the study of GCL can be categorized into three key aspects: data augmentation, contrast level, and negative sample sampling strategies.

2.2.1. Data Augmentation Methods. Data augmentation methods not only enhance the robustness and generalization ability of graph contrastive learning but also enable the model to capture the underlying patterns and relationships within the graph more effectively. Early research efforts focused on designing heuristic algorithms [32–35] to augment graphs. Later, Qiu et al. [32] employ stochastic subgraph sampling as a data enhancement strategy, combined with contrastive learning, to learn intrinsic and migratable structural representations. Zhu et al. [34] proactively select Dropedge as a data enhancement strategy and supplement it with node centrality to compensate for randomly deleted edges. By integrating such strategies with contrastive learning, researchers can achieve more robust and accurate models that can effectively capture the complex patterns and relationships present with fewer graphs. Yang et al. [36] augmented the collaborative filtering paradigm with two adaptive contrasting view generators, a graph generation model and a graph denoising model. Mei et al. [37] used variational graph reconstruction to estimate Gaussian distributions for nodes and based on the learned distribution to generate comparison views.

2.2.2. Contrastive Levels. Contrastive levels determine the hierarchy at which graph elements are compared, encompassing different levels such as node-to-node, graph-to-graph, and node-to-graph comparisons. You et al. [35] put forth a model within the graph-graph contrastive learning category, which aims to generate graph representations that exhibit similar or improved levels of generalization, transferability, and robustness. Zhu et al. [34] prioritize node-node contrastive learning and employ a strategy of introducing supplementary noise to unimportant node features, intentionally corrupting the representations of those nodes. Hassani et al. [33] employ node-graph contrastive learning to acquire richer representations, facilitating separate node and graph classification tasks. By considering these varying levels, GCL can effectively capture diverse aspects of the graph structure and interactions, leading to a more comprehensive analysis and understanding of the data.

2.2.3. Negative Samples Sampling. Negative samples sampling plays a critical role in optimizing the contrastive learning process. The negative sample sampling strategy determines how dissimilar graph element pairs are chosen as negative samples, allowing the model to effectively discriminate between similar and dissimilar graph elements. Early studies [38–41] initially utilized the removal of positive samples as negative samples, but this approach often led to false negatives. To address this challenge, researchers proposed alternative strategies, such as combining clustering [42] or curriculum learning [43], for sample selection. Additionally, some studies [44–46] introduced the concept of no-negative samples, which yielded promising results in enhancing the effectiveness of contrastive learning.

These techniques improve contrastive learning by expanding graph data, and identifying suitable levels of comparison, and selecting pertinent negative samples. This comprehensive approach enhances the accuracy and performance of rumor detection systems, facilitating a profound comprehension of graph structure and dynamics.

3. Preliminaries

Suppose we have a rumor detection dataset $\mathscr{M} = \{m^i | i \in [1, |\mathscr{M}|]\}$, where m^i represents the i-th event. Each event consists of a source post p_0 and a set of relevant forward posts $P^i = \{p_1, p_2, \cdots\}$. For each event m_i, according to its diffusion threads, we can construct a propagation graph $G_i = \{\mathscr{V}^i, \mathscr{E}^i\}$, where $\mathscr{V}^i = \{p_0, p_1, \ldots, p_{|\mathscr{V}^i|-1}\}$ and $\mathscr{E}^i = \{e_{jk} | j, k \in \mathscr{V}^i\}$ is the corresponding directed edge list, where $e_{jk} = 1$ if p_k is a comment or retweet of p_j, otherwise, $e_{jk} = 0$. Although our work mainly focuses on analyzing the potential commonality of propagation structures, to achieve better rumor detection performance, we compute extra TF-IDF values for each post within m_i to form the initial node feature matrix, which is denoted as $\mathbf{X}_i = \{\mathbf{x}_i^0, \mathbf{x}_i^1, \ldots, \mathbf{x}_i^{|\mathscr{V}^i|-1}\} \in \mathbb{R}^{|\mathscr{V}^i| \times d}$, where d is the dimensionality of node feature.

In this paper, the rumor detection task aims at training a model to allocate label $\hat{y}$ for a specific event m_i, which can be regarded as a multiclassification problem. Here, $\hat{y}$ takes one of the four finer-grained classes: NR, FR, TR, and UR (i.e., nonrumor, false rumor, true rumor, and unverified rumor).

4. Methods

In this section, we introduce our rumor detection model–UICL (Uncertainty-Inference Contrastive Learning-based Rumor Detection model), which first utilizes contrastive learning, along with edge-inference and edge-dropping augmentation techniques, to train an efficient graph encoder in an unsupervised manner while also preserving the authenticity of the rumor propagation structure. Then, UICL further improves model performance in the rumor detection task by fine-tuning with labeled data. Figure 1 shows the overall framework of UICL, which mainly consists of two stages: (1) contrastive pretraining and (2) model fine-tuning. We will describe these in detail in the following subsections.

4.1. Propagation Graph Encoder. To extract structural information from the propagation graph, we are inspired by the recent success of graph neural networks (GNNs) in unstructured data learning. Specifically, in our experiments, we utilize multiple GNN layers as our propagation graph encoder (GEN), and a single GNN layer can be defined as

$$\mathbf{H}^{(k)} = \mathrm{MLP}\left(\widehat{\mathbf{A}}\mathbf{H}^{(k-1)}\right), \tag{1}$$

where $\widehat{\mathbf{A}}$ and $\mathbf{H}^{(k-1)} = \{\mathbf{h}_0^{(k-1)}, \mathbf{h}_1^{(k-1)}, \ldots, \mathbf{h}_i^{(k-1)}, \ldots, \mathbf{h}_{|\mathscr{V}|-1}^{(k-1)}\}$ represent the normalized adjacency matrix of $\mathbf{A}$ and node features at k-th layer, respectively. $\mathrm{MLP}(\cdot)$ is a multilayer perceptron with nonlinear activation function. The initial node features are $\mathbf{H}^{(0)} = \mathrm{MLP}(\mathbf{X})$, and the final node embedding $\mathbf{H}$ is the output of the last GNN layer $\mathbf{H}^{(k)}$ in GEN. To obtain the global representation of the propagation graph $\mathbf{h}^G$, we apply a mean-pooling READOUT operation on $\mathbf{H}$:

$$\mathbf{h}^G = \mathrm{MEAN}(\mathbf{H}). \tag{2}$$

Note that the output of GEN can be either node representation $\mathbf{H}$ or graph representation $\mathbf{h}^G$, depending on what we need to extract local or global information from the propagation graph of an event.

4.2. Edge-Wise Augmentation. As we introduced in Section 2.1, the main idea behind contrastive learning is to maximize mutual information (MI) between positive samples, while minimizing it among negative samples [47]. Specifically, the positive samples of an instance are generated by augmenting the original instances. Existing graph augmentation methods either randomly remove nodes/edges, or randomly mask node/edge features in graphs [13], which will destroy the semantics of the propagation graph. Furthermore, the "echo chamber effect" and unreliable relationships can exist in the propagation graph due to the deceptive tactics of rumor producers, as well as the limited social context data available. To overcome these challenges, we propose two edge-wise augmentation strategies: (1) an edge-inference augmentation module to overcome the uncertainty issue by considering edges' importance in the propagation graph, in this work, we define the edge importance as its weight in a graph. And (2) an edge-drop augmentation module aims at alleviating the echo chamber effect by randomly removing a portion of edges from the propagation graph.

4.2.1. Edge-Inference Augmentation. For a propagation graph G, we first defined an intensity function $\mathrm{Int}(\cdot)$ to compute the intensity score $\mathbf{r}^{ij}$ of edge $e_{ij} \in \mathscr{E}$ between two nodes v_i and v_j, which is calculated as

$$\mathbf{r}^{ij} = \mathrm{Int}\left(\left\|\mathbf{h}_i - \mathbf{h}_j\right\|\right), \tag{3}$$

where $\mathrm{Int}(\cdot)$ can be any type of nonlinear neural network (in our experiments, we employ a convolutional layer as $\mathrm{Int}(\cdot)$). $\mathbf{h}_i$ and $\mathbf{h}_j$ are embeddings of nodes v_i and v_j, respectively, which are calculated through feeding the original node feature matrix $\mathbf{X}$ to GEN layer. Assume we have T types of different relations, such as friends, followers, and

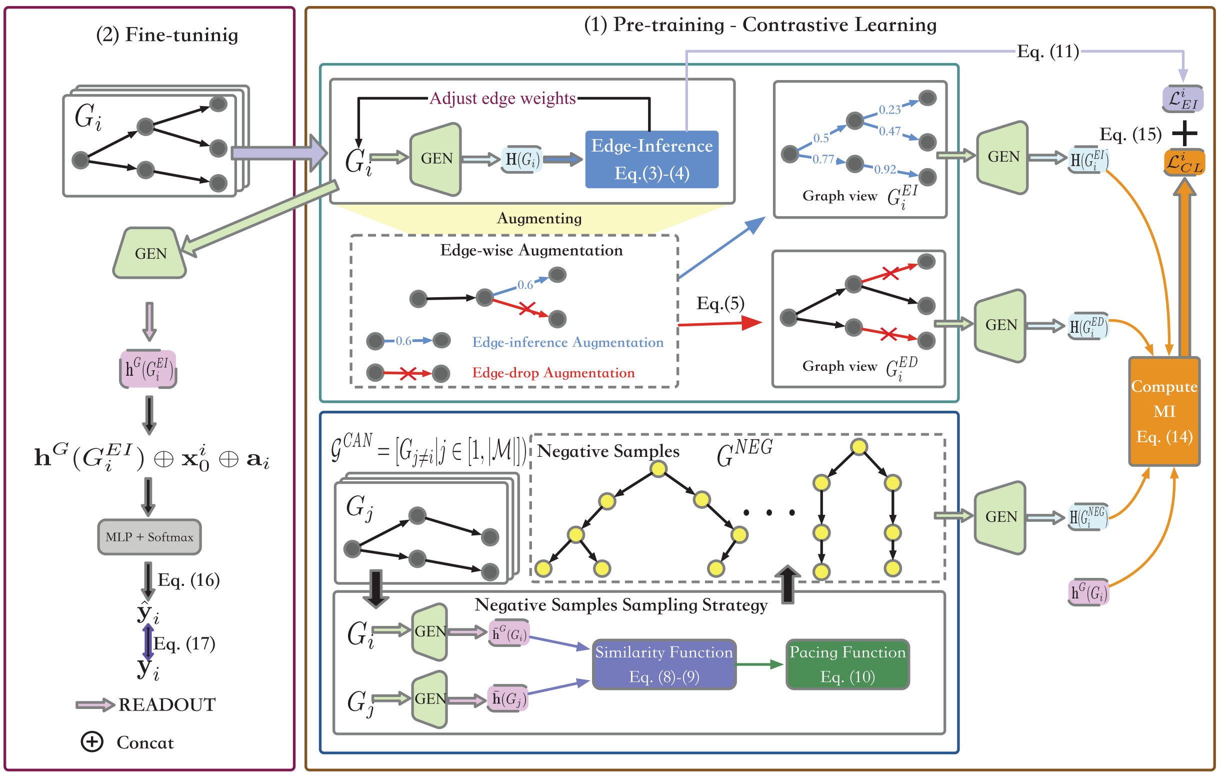


Figure 1: Overview of UICL. UICL consists of two main parts: (1) pretraining via edge-wise augmentation contrastive learning and (2) fine-tuning with labeled data.

forwarding-only. Then, the corresponding weighted adjacency matrix $\mathbf{A}^{\mathrm{EI}}$ is updated as follows:

$$\mathbf{A}^{\mathrm{EI}}=\sum_{t=1}^{T}\sigma\left(\mathbf{W}_t\mathbf{R}+\mathbf{b}_t\right)\cdot\mathbf{A}, \tag{4}$$

where $\mathbf{W}_t$ and $\mathbf{b}_t$ are learnable parameters for the relation t, $\mathbf{R}=\{\mathbf{r}^{ij}|(i,j)\in\mathscr{E}\}$ is the intensity score matrix, and $\mathbf{A}^{\mathrm{EI}}$ and $\mathbf{A}$ are updated and original adjacency matrices, respectively. σ refers to Sigmoid$(\cdot)$. After the edge-inference augmentation module, we denote the augmented graph as G^{EI}.

4.2.2. DropEdge. Suppose the dropping rate is p_{drop}, the original adjacency matrix $\mathbf{A}$ after edge-drop augmentation is

$$\mathbf{A}^{\mathrm{ED}}=\mathbf{A}-\mathbf{A}_{\mathrm{drop}}, \tag{5}$$

where $\mathbf{A}_{\mathrm{drop}}$ is contrasted by randomly sampling $|\mathscr{E}|\times p_{dr}$ edges from the original edge set. The created edge-drop view is represented as G^{ED}.

4.3. Negative Samples. Most of the existing graph contrastive learning methods select negative samples using random strategies [13], which are not appropriate for our rumor detection task since the complexity of the propagation structure for different events varies [47]. Events with a high forwarding activity frequency may have more complex propagation structures. In addition, tweets that share the same label may not necessarily discuss the same events or themes. Therefore, we employ a curriculum learning-inspired strategy for sampling negative samples [43]. Specifically, we introduce a Similarity Function and a Pacing Function to select negative samples for training from a list of candidates.

4.3.1. Similarity Function. For a propagation graph G_i in a training batch B, we place the rest of the data except itself as its candidate negative samples $\mathscr{G}_i^{\mathrm{CAN}}=\left\{G_{j\neq i},\, j\in[1,|B|]\right\}$. We first feed all candidate negative samples into GEN followed by an MLP projection layer and output a set of candidate representations – $\mathbf{H}_{\mathscr{G}_i^{\mathrm{CAN}}}=\left\{\overline{\mathbf{h}}_{j\neq i}^G,\, j\in[1,|B|]\right\}$. Specifically, for each candidate, we have

$$\mathbf{h}_j^G=\mathrm{GEN}\left(G_j\right), \tag{6}$$

$$\overline{\mathbf{h}}_j^G=\mathrm{MLP}\left(\mathbf{h}_j^G\right). \tag{7}$$

Then, we utilize a cosine similarity function to measure the difference between each candidate negative sample and the original graph G_i:

$$\mathrm{Sim}\left(G_i,\mathscr{G}_i^{\mathrm{CAN}}\right)=\left\{\cos\left(\overline{\mathbf{h}}_{G_i}^G,\overline{\mathbf{h}}_{G_j}^G\right),G_j\in\mathscr{G}_i^{\mathrm{CAN}}\right\}. \tag{8}$$

The cosine similarity between $\overline{\mathbf{h}}_{G_i}^G$ and $\overline{\mathbf{h}}_{G_j}^G$ is computed as follows:

$$\cos\left(\overline{\mathbf{h}}_{G_i}^{G}, \overline{\mathbf{h}}_{G_j}^{G}\right) = \frac{\left|\overline{\mathbf{h}}_{G_j}^{G} \cdot \overline{\mathbf{h}}_{G_i}^{G}\right|}{\left|\overline{\mathbf{h}}_{G_j}^{G}\right| \cdot \left|\overline{\mathbf{h}}_{G_i}^{G}\right|}. \tag{9}$$

Usually, the higher the cosine similarity value, the less the difference.

4.3.2. Pacing Function. After calculating the similarity score between G_i and all candidate negative samples, we sort candidates by their similarity score in ascending order. Then, we employ a pacing function [43] $s(t)$ to select the negative samples with the lowest similarity scores. More specifically, the pacing function is used to specify the number of negative samples at a specific epoch t, which is defined as follows:

$$s(t) = \left(\frac{t}{C}\right)^{\lambda} \cdot K, \tag{10}$$

where K and C are the number of candidates and training epochs, respectively. λ is an adjusting parameter, which can be $1/2, 1, 2$. Finally, at epoch t, the list of negative samples consists of $s(t)$ lowest scored samples from the candidates.

4.4. Pretraining with Contrastive Learning. In this section, we present the details of how UICL completed the pretraining phase, starting with introducing the training loss function, followed by a concise description of the training procedure. In UICL, we propose two distinct yet jointly trained loss functions: one derived from the edge-inference augmentation module and another employed for propagation graph contrastive learning.

4.4.1. Unsupervised Edge-Inference Loss. We were inspired by the work [48] and utilized the unsupervised learning loss $\mathscr{L}_{\mathrm{EI}}$ to train the edge-inference augmentation module:

$$\mathscr{L}_{\mathrm{EI}}^{i} = \mathbb{E}\left[D_{\mathrm{KL}}\left(p\left(\mathbf{A}_i^{\mathrm{EI}} \middle| \mathbf{R}_i\right) \middle\| q\left(\mathbf{A}_i^{\mathrm{GAU}} \middle| \mathbf{R}_i\right)\right)\right], \tag{11}$$

where $\mathbf{A}^{\mathrm{GAU}}$ is Gaussian sampling edge weights sampled from

$$\mathbf{A}_i^{\mathrm{GAU}} = \mathscr{N}\left(\mu_i, \delta_i^2\right), \tag{12}$$

$$\mu_i = f_{\mu}(\mathbf{R}_i), \quad \delta_i^2 = f_{\delta}(\mathbf{R}_i), \tag{13}$$

where μ and δ^2 are means and variances, respectively. f_{μ} and f_{δ} are transformation layers. Note that, we model the prior distribution of each latent relation t independently.

4.4.2. Contrastive Learning Loss. We accomplish contrastive learning by maximizing the mutual information [47] over the rumor propagation graph dataset, which is computed as

$$\begin{aligned}\mathscr{L}_{CL}^{i} = {} & \mathbb{E}\left[-\log\left(1 + e^{-F\left(\mathbf{H}\left(G_i^{\mathrm{pos}}\right), \mathbf{h}^{G}\left(G_i\right)\right)}\right)\right] \\ & - \mathbb{E}\left[\log\left(1 + e^{F\left(\mathbf{H}\left(G_i^{neg}\right), \mathbf{h}^{G}\left(G_i\right)\right)}\right)\right],\end{aligned} \tag{14}$$

where F is the discriminator (in our implementation, we compute the dot product between vectors, which is a standard method for measuring similarity in machine learning). G_i^{pos} and G_i^{neg} are the positive and negative samples of input event graph sample G_i, respectively.

4.4.3. Total Loss. Then, the final loss function in the contrastive learning part is defined as a summation of the unsupervised loss $\mathscr{L}_{\mathrm{EI}}$ and the contrastive loss $\mathscr{L}_{\mathrm{CL}}$, i.e.,

$$\mathscr{L} = \frac{1}{B}\sum_{i=1}^{B} (1-\alpha)\mathscr{L}_{\mathrm{EI}}^{i} + \alpha\mathscr{L}_{\mathrm{CL}}^{i}, \tag{15}$$

where B is the batch size, α is a balance parameter, which is used to control the contribution of each loss term, and can be adjusted manually. The whole procedure of the pretraining phase is shown in Algorithm 1.

4.5. Rumor Detection with Fine-Tuning. In the fine-tuning step, we first initialize the model's parameters with the pretrained parameters from the contrastive learning phase, i.e., GEN and the edge-inference augmentation module. Then, we train the detection module using labeled data. Specifically, for a given event m^i, we obtain its adjusted weight graph G^{EI} from the pretrained edge-inference augmentation module. We then compute the event representation by feeding G^{EI} into the pretrained GEN. The acquired event representation is denoted as $\mathbf{h}^{G}(G_i^{\mathrm{EI}})$. In addition, we also calculate the averaged textual features $\mathbf{a}_i$ for m^i by averaging the original features of all related posts P, i.e., $\mathbf{a}_i = (1/|\mathscr{V}^i|)\sum_{j=0}^{|\mathscr{V}^i|-1}\mathbf{x}_j$. Finally, we concatenate $\mathbf{h}^{G}(G_i^{\mathrm{EI}})$, $\mathbf{a}_i$, and the source tweet feature $\mathbf{x}_0^i$ to form the input to feed into the prediction module. The prediction module consists of an MLP layer with a softmax activation function.

$$\widehat{\mathbf{y}}_i = \mathrm{Softmax}\left(\mathrm{MLP}\left(\mathbf{h}^{G}\left(G_i^{\mathrm{EI}}\right) \oplus \mathbf{x}_0^i \oplus \mathbf{a}_i\right)\right), \tag{16}$$

where $\oplus$ is a concatenate operation. We calculate the cross-entropy loss between the predictions $\widehat{\mathbf{Y}}$ and ground truth $\mathbf{Y}$ for all events involved in the dataset, which is defined as

$$\mathscr{L}(\mathbf{Y}, \widehat{\mathbf{Y}}) = -\frac{1}{B}\sum_{i=1}^{B} \mathbf{y}_i \log \widehat{\mathbf{y}}_i + \lambda\|\Theta\|_2^2, \tag{17}$$

where $\|\Theta\|_2^2$ is the L_2 regularizer over all the model parameters Θ and λ is the trade-off coefficient. In this work, we use Adam [49] as optimizer. The fine-tuning algorithm of our UICL model is summarized in Algorithm 2.

5. Experiments

In this section, we first introduce the datasets used in experiments. Then, we give a brief introduction to the related experimental settings. Finally, we provide a detailed analysis of the experimental results. To explain the experimental results more clearly, we would aim to provide a quantitative characterization of the following research-related questions:

Inputs: A set of propagation graphs $\mathcal{G}$, where each $G_i \in \mathcal{G}$ corresponds to an initial node feature matrix $\mathbf{X}_i$, the number of relation t, edge dropping rate d_p, propagation graph encoder-GEN, and edge-inference augmentation module-EIA.
Outputs: Trained model parameters θ_{GEN} and θ_{EIA} for GEN and EIA, respectively.
(1) Initialize θ_{GEN} and θ_{EIA} with random weight values;
(2) **for** epoch from 1 to maxEpoch **do**
(3) **for** each mini-batch $\{G_1, G_2, \ldots, G_B\}$ of $\mathcal{G}$ **do**
(4) **for** each G_i in the mini-batch **do**
(5) Store other graphs from the same mini-batch as G_i's candidate negative samples $\mathcal{G}_i^{\text{CAN}}$;
(6) Calculate local node representation $\mathbf{H}(G_i)$ and global graph representation $\mathbf{h}^G(G_i)$ based on equations (1) and (2), respectively;
(7) Generate two augmented views G^{EI} and G^{ED} based on equations (3)–(5), respectively;
(8) Compute propagation graph representation $\mathbf{H}(G_i^{\text{EI}})$ and $\mathbf{H}(G_i^{\text{ED}})$ based on equation (1);
(9) Compute global graph representation for all candidates from $\mathcal{G}_i^{\text{CAN}}$ based on equations (6) and (7);
(10) Generate negative samples G_i^{neg} based on equations (8)–(10);
(11) Compute unsupervised edge-inference loss $\mathcal{L}_{\text{EI}}^i$ for G_i via equation (11);
(12) Compute contrastive learning loss $\mathcal{L}_{\text{CL}}^i$ for G_i via equation (14);
(13) **end for**
(14) Compute total loss $\mathcal{L}$ via equation (15);
(15) Update θ_{GEN} and θ_{EIA} to minimize loss $\mathcal{L}$;
(16) **end for**
(17) **end for**
(18) **return** Trained model parameters θ_{GEN} and θ_{EIA}.

Algorithm 1: Pretraining contrastive learning procedure.

Inputs: A set of event propagation graphs $\mathcal{G}$, where each $G_i \in \mathcal{G}$ corresponds to an initial node feature matrix $\mathbf{X}_i$ and ground truth label $\mathbf{y}_i$; GEN; EAI; rumor prediction layer-RPL.
Outputs: Trained model parameters θ_{GEN}, θ_{EAI}, and θ_{RPL}.
(1) Initialize θ_{GEN}, θ_{EAI} with the pretraining parameters, and θ_{RPL} with random weight values;
(2) **for** epoch from 1 to maxEpoch **do**
(3) **for** each mini-batch of $\mathcal{G}$ **do**
(4) **for** each graph G_i in mini-batch **do**
(5) Generate edge-inference graph G_i^{EI} via EAI.
(6) Compute event representation $\mathbf{h}^G(G_i^{\text{EI}})$ via feeding G_i^{EI}, $\mathbf{X}_i$ to GEN;
(7) Calculate averaged node feature $\mathbf{a}_i = 1/|\mathcal{V}^i| \sum_{j=0}^{|\mathcal{V}^i|-1} \mathbf{x}_j$;
(8) Concatenate $\mathbf{h}^G(G_i^{\text{EI}})$, $\mathbf{x}_0^i$ and $\mathbf{a}_i$, and feed it to RPL and make prediction $\widehat{\mathbf{y}}_i$ based on equation (16);
(9) **end for**
(10) Calculate the cross-entropy loss $\mathcal{L}(\mathbf{Y}, \widehat{\mathbf{Y}})$ using equation (17)
(11) Update θ_{GEN}, θ_{EAI} and θ_{RPL} to minimize loss $\mathcal{L}(\mathbf{Y}, \widehat{\mathbf{Y}})$;
(12) **end for**
(13) **end for**
(14) **return** Trained model parameters θ_{GEN}, θ_{EAI} and θ_{RPL}.

Algorithm 2: Fine-tuning procedure.

RQ1. How does UICL perform on rumor detection compared with the state-of-the-art baselines?

RQ2. What is the effect of each component of UICL?

RQ3. Can UICL detect rumors in the early stages?

5.1. Experimental Settings

5.1.1. Datasets. To assess the effectiveness of UICL, we conduct our experiments on three publicly available real-world datasets: Twitter15, Twitter16 [50], and Weibo [7]. All of these datasets are derived from existing mainstream online social networks, i.e., Twitter in the US and Weibo in China. Specifically, Twitter15 and Twitter16 were collected in 2015 and 2016, respectively. Each event in these datasets comprises a source tweet and its corresponding series of subsequent retweets, replies, or social engagements. Besides that, each event is also associated with a specific label from the following four categories: nonrumors, false rumors, unverified rumors, and true rumors. Similar to Twitter datasets, the Weibo dataset comprises the original posts of specific events and all their retweets/replies obtained through the Weibo API. The only difference is that in the Weibo dataset, there only exist rumors and nonrumors. Table 1 shows the statistics of three datasets.

5.1.2. Baselines. We compare our UICL with the following state-of-the-art baselines, including

(i) GRU [7]: A gated recurrent unit-based rumor detection model that utilizes recurrent neural networks to understand the sequential impact of tweets over time and extract high-level features from relevant posts.

(ii) RvNN [9]: A neural rumor detection framework employing tree-structured recursive neural networks (RvNN) is introduced. It seeks to correlate content semantics and propagation cues, enhancing the detection of rumors in microblog posts.

(iii) PPC [17]: A model that integrates a recurrent neural network (RNN) and convolutional neural network (CNN) to effectively capture global and local variations in user characteristics along the propagation paths and further use the learned features to detect rumors.

(iv) StA-PLAN [19]: A postlevel rumor detection model, which utilizes multitask learning and attention mechanisms to concurrently predict both the veracity and the type of an event.

(v) BiGCN [10]: Captures the patterns of rumor propagation and the structures of wide dispersion through its dual operation on both top-down and bottom-up rumor propagation processes via bidirectional GCN.

(vi) RDEA [13]: RDEA is the first rumor detection model that attempts to employ contrastive learning for rumor detection based on propagation graphs.

(vii) CCFD [51]: Utilizes curriculum learning techniques to improve negative sample sampling in contrastive learning-based rumor detection.

(viii) GACL [28]: A rumor detection model that integrates both contrastive learning and adversarial learning mechanisms to capture event-invariant features for effective rumor detection.

5.1.3. Evaluation Metrics. Different evaluation protocols are employed to assess the model's performance on Twitter and Weibo datasets. For Twitter datasets, we report accuracy (ACC) over four classes, along with F-measure (F_1) calculated for each class. As for the Weibo dataset, we evaluate the ACC, F_1, precision, and recall results over the two categories.

5.1.4. Implementation Details. The proposed UICL model is implemented using PyTorch (https://pytorch.org/). Following previous work [10], we randomly divide the datasets into five parts and conduct 5-fold cross-validation to ensure the reliability of the results. In our experiments, the dimension value (d) of the word embedding vector is set to 5000. We employ a 2-layer Graph Convolutional Network (GCN) with both hidden and output features of each node set to 64. The learning rate is established at 0.0005. We define the number of latent relations (T) as 2. To enhance model performance, the DropEdge technique is applied with a drop rate of 0.2. The training involves 200 epochs, each comprising 100 iterative updates. However, training may be stopped earlier if the validation loss fails to decrease after 10 epochs. Moreover, we set the weight decay coefficient to $1e-4$ and the dropout rate is 0.2.

5.2. Overall Performance (RQ1). Tables 2, 3, and 4 report the performance comparison among UICL and baselines on three datasets, from which we have the following observations:

(O1) GRU exhibits the worst performance among all baselines based on the results across all datasets. This is attributed to its only focus on extracting textural features and ignoring other important features, such as diffusion features and user profiles. RvNN and PPC outperform GRU, which demonstrates the importance of propagation structure and temporal information for rumor detection. Although PLAN employs transformer architecture to augment the model's capacity in extracting textual features, its performance improvement over RvNN and PPC is insignificant due to the neglect of the structural features inherent in rumor propagation.

(O2) BiGCN, RDEA, CCFD, and GACL, these GNN-based models outperform other baselines owing to the advantage of GNNs in extracting structural features from the propagation graph. In addition, the results further prove the importance of potential information about the propagation structure for rumor detection. While RDEA utilizes BiGCN as its backbone, it markedly outperforms BiGCN. This improvement is achieved by incorporating contrastive learning into the BiGCN framework, which alleviates the influence of limited labeled data. CCFD outperforms RDEA by introducing a curriculum learning-based negative sample sampling strategy. Besides that, GACL, in turn, demonstrates superior performance over CCFD, due to its success in the incorporation of adversarial learning. This enables the modeling of sociological principles inherent in real social networks, coupled with the utilization of labeled explicit selection methods for sampling negative instances.

(O3) Our proposed model, UICL, achieves the best performance across a majority of metrics evaluated on the three datasets. In contrast to the top-performing baselines, i.e., CCFD and GACL, while all three models show competitive performance, UICL consistently achieves a higher detection accuracy than both CCFD and GACL. This is because our UICL not only mitigates label influence through the incorporation of contrastive

Table 1: Statistics of the datasets.

Description	Twitter15	Twitter16	Weibo
Events #	1,490	818	4,664
Nonrumors #	374	205	2,351
False rumors #	370	205	—
Unverified rumors #	374	203	—
True rumors (rumors) #	372	205	2,351
Posts #	331,612	204,820	3,805,656
Avg. # Posts of event	223	251	816
Max # posts of event	1,768	2,765	59,318
min # posts of event	55	81	10

Table 2: Overall performance comparison of rumor detection on Twitter15.

Method	Acc.	F1			
		NR	FR	UR	TR
GRU	0.641	0.684	0.634	0.571	0.688
RvNN	0.723	0.682	0.758	0.654	0.821
PPC	0.697	0.689	0.760	0.696	0.645
PLAN	0.787	0.775	0.807	0.775	0.768
BiGCN	0.836	0.791	0.842	0.801	0.887
RDEA	0.855	0.831	0.857	**0.903**	0.816
CCFD	0.856	*0.848*	0.861	0.816	**0.893**
GACL	*0.866*	0.820	**0.898**	*0.843*	0.837
UICL	**0.868**	**0.882**	*0.878*	0.831	*0.890*

"NR": nonrumor; "FR": false rumor; "UR": unverified rumor; "TR": true rumor. The best method is shown in bold, and the second best is shown as italic.

Table 3: Overall performance comparison of rumor detection on Twitter16.

Method	Acc.	F1			
		NR	FR	UR	TR
GRU	0.636	0.617	0.715	0.527	0.577
RvNN	0.737	0.662	0.743	0.708	0.835
PPC	0.702	0.608	0.711	0.664	0.816
PLAN	0.799	0.754	0.821	0.779	0.836
BiGCN	0.864	0.788	0.859	0.864	0.886
RDEA	0.880	0.823	0.878	0.875	0.927
CCFD	0.886	0.819	**0.884**	*0.892*	**0.954**
GACL	*0.896*	*0.862*	0.869	0.886	0.926
UICL	**0.907**	**0.858**	*0.882*	**0.933**	*0.934*

"NR": nonrumor; "FR": false rumor; "UR": unverified rumor; "TR": true rumor. The best method is shown in bold, and the second best is shown as italic.

Table 4: Overall performance comparison of rumor detection on Weibo.

Method	Acc.	Prec.	Rec.	*F*1
GRU	0.732	0.738	0.715	0.726
RvNN	0.788	0.819	0.746	0.737
PPC	0.765	0.746	0.781	0.762
PLAN	0.857	0.842	0.881	0.861
BiGCN	0.867	0.870	0.860	0.865
RDEA	0.867	0.870	0.860	0.865
CCFD	0.892	*0.880*	0.901	0.890
GACL	*0.910*	0.876	**0.956**	0.914
UICL	**0.920**	**0.915**	*0.918*	**0.916**

The best method is shown in bold, and the second best is shown as italic.

learning but also considers edge uncertainty in the propagation graph. From the results of F_1 metrics, CCFD excels in detecting nonrumors, while UICL demonstrates superior performance in identifying the rest types of rumors, i.e., false rumor, unverified rumor, and true rumor. However, UICL achieves the highest F_1 score in identifying true rumors within the Twitter16 dataset. On the Weibo dataset, UICL distinguishes itself with the highest accuracy, precision, and a robust overall F_1 score. In summary, UICL demonstrates outstanding performance across various metrics, demonstrating its efficacy in rumor detection on Online Social Networks.

(O4) UICL demonstrates markedly improved performance on Weibo as opposed to the results on Twitter15 and Twitter16. Several potential factors account for this discrepancy: (1) a substantial portion of posts in the Twitter15 and Twitter16 datasets lack textual content, resulting in limited input features. (2) The detection task on the Weibo dataset can be regarded as a binary classification task, which is naturally easier compared to multiclass classification tasks using the same inputs.

5.3. Ablation Study (RQ2). In this section, we conduct extensive ablation studies to evaluate the effectiveness of each component in the UICL model, including propagation graph augmentation and negative mining strategies, as well as the balance value α in the loss function. We conclude with a discussion on UICL's detection performance when trained on limited labeled data.

5.3.1. Augmentation Strategies. Specifically, we derive the following ten variants of UICL with different augmentation strategies:

(i) D: In variant *D*, only the Edge-drop augmentation in UICL is used to generate the positive view.

(ii) I: In variant *I*, we only retain edge-inference augmentation and deactivate Edge-drop augmentation.

(iii) SI: Variant SI substitutes UICL's Dropedge augmentation with subgraph augmentation, while retaining edge-inference augmentation.

(iv) NI: In variant NI, node masking replaces Dropedge augmentation, with edge-inference augmentation remaining intact.

(v) SD: Variant SD employs subgraph augmentation in place of edge-inference augmentation in UICL, yet maintains Dropedge augmentation.

(vi) ND: For ND, node masking supersedes edge-inference augmentation, while keeping Dropedge augmentation in effect.

(vii) SN: Variant SN integrates both subgraph augmentation and node masking, replacing UICL's edge-wise augmentation strategy.

(viii) SDI: SDI enhances UICL's edge-wise augmentation strategy with an additional subgraph augmentation.

(ix) NDI: In NDI, an additional node masking method is combined with the original edge-wise augmentation strategy in UICL.

(x) SNDI: Variant SNDI amalgamates node masking and subgraph augmentation with UICL's original edge-wise augmentation strategy.

In Figure 2, we visually explain different augmentation strategies mentioned in variants D to SNDI. The performance of the original UICL and its variants, each employing different augmentation strategies, is depicted in Figure 3, from which we can observe that (O1) UICL outperforms all other variants. (O2) The results of *I* and *D* reveal that single view augmentation improves detection performance to some degree when compared to the state-of-the-art propagation-based method–BiGCN. However, it remains significantly behind variants that employ various augmentation strategies to generate multiple views of the original graph, which possess stronger representation learning capabilities. (O3) The performance of NI and ND marginally surpasses that of SI and SD, respectively. This can be attributed to the more comprehensive structural features introduced by node masking, whereas subgraph augmentation, by focusing on localized subsets of the graph, potentially neglects important global structural contexts. (O4) Comparing SDI, NDI, and SNDI with our UICL, we observe that edge-wise augmentation is sufficiently efficient for modeling the underlying structure of the propagation graph. However, incorporating additional augmentation strategies can lead to a decrease in model performance. Especially, SNDI performs the worst among these methods. This decline is attributed to the introduction of noise into the propagation structure and an increase in the model's computational complexity. (O5) From the results of SI, NI, and UICL, we can find that among the various combinations, Dropedge is more compatible with edge-inference augmentation. This compatibility is due to Dropedge's ability to capture effective features from the potentially diverse network structures, as well as both of these augmentation methods operating on edges.

5.3.2. Negative Mining Strategies. The way to sample negative samples affects both the training speed and detection accuracy of the model [52]. To demonstrate the effectiveness of the negative mining strategy in UICL, we derived two variants NR and NA. The NR randomly selects a portion of propagation graphs from other events within the same batch as negative samples, whereas NA treats all propagation graphs from other events in the same batch as negative samples. The experimental results are shown in Figure 4. We observe that our UICL, which incorporates a curriculum learning-inspired strategy for sampling negative samples, not only outperforms the other two variants in detection accuracy but also significantly reduces training time. This demonstrates that the introduction of the similarity function

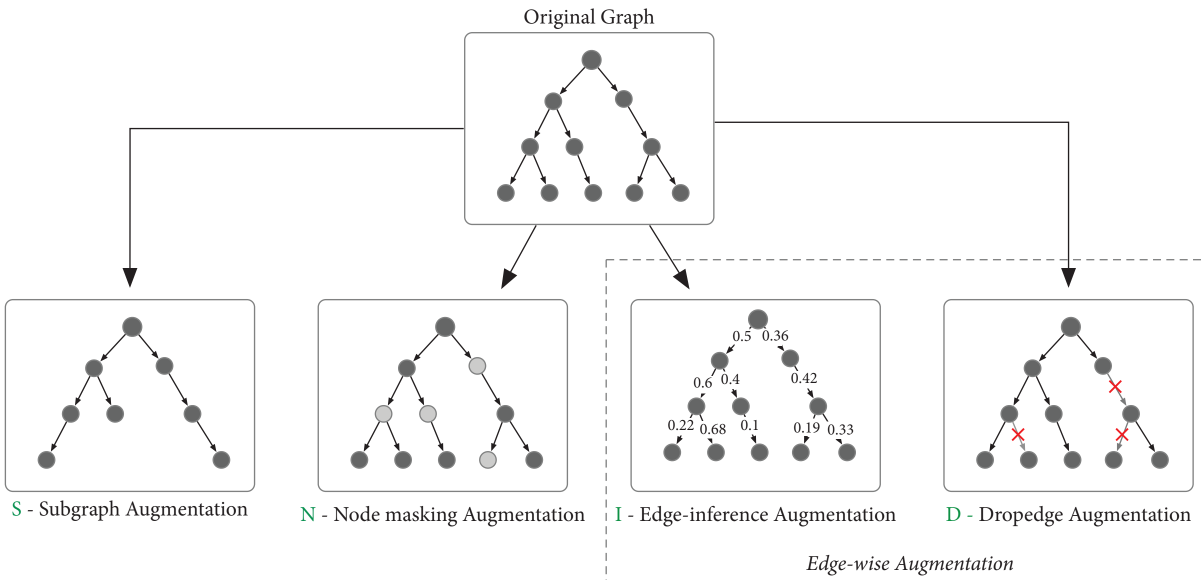


FIGURE 2: Augmentation strategies in variants.

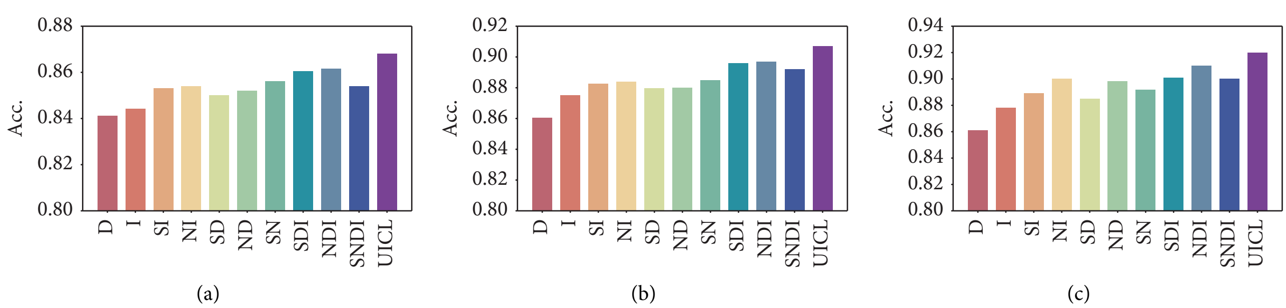


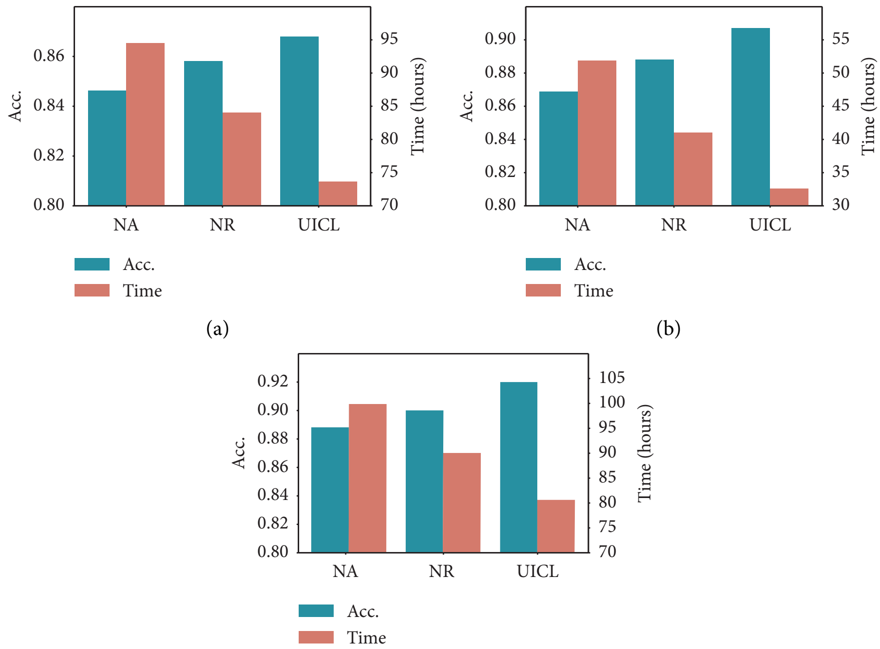
FIGURE 3: Ablation studies on different augmentation strategies. (a) Twitter15. (b) Twitter16. (c) Weibo.

and pacing function can effectively select the most discriminative negative samples, thereby enhancing the model's representation learning ability, as well as reducing the number of required samples. Additionally, comparing NA to NR, we find that introducing more negative samples fails to enhance the model's performance and significantly increases the model's training time.

5.3.3. Model Sensitivity to α. We conduct experiments using varying values of α across three datasets, with α ranging from 0 to 1. The results are illustrated in Figure 5. We can see that the optimal value for α across all datasets is 0.6. UICL's performance initially improves with increasing values of α, gradually reaching saturation after 0.2. Beyond 0.6, the performance slightly decreases.

5.3.4. Data Imbalance. Data imbalance is a common yet unresolved problem. To demonstrate our model's superiority in addressing this issue, we conducted experiments with Limited Labeled Data during the fine-tuning phase, and the results are presented in Figure 6. The performance of the models improves as the amount of labeled data increases. In cases where the labeled dataset is small, all models exhibit relatively lower performance. However, the UICL model maintains a significant advantage, which can be potentially attributed to its ability to capture structural uncertainty features. Furthermore, it is observed that labeled data has the most significant impact on the performance of the GACL model. With only a 5% labeled dataset, the detection accuracies for the Twitter15, Twitter16, and Weibo datasets are approximately 41.3%, 37.4%, and 69.2%, respectively. However, when the labeled dataset is increased to 20%, the corresponding detection accuracies improve to 63.3%. This pattern demonstrates that the GACL model heavily relies on labeled data, particularly in the sampling phase of positive and negative samples within its supervised comparative learning framework, where insufficient labeled data can limit the model's ability to learn potential features in negative samples, resulting in lower detection accuracy.

5.4. Early Detection (RQ3). Detecting rumors at an early stage is crucial to hinder the spreading of false information and to mitigate the harmful impact on individuals. Here, we select the following detection deadlines, i.e., 5, 20, 30, . . ., 90 minutes. Specifically, we utilize the posts published before



(a) (b) (c)

Figure 4: Performance comparison of different negative samples sampling methods. (a) Twitter15. (b) Twitter16. (c) Weibo.

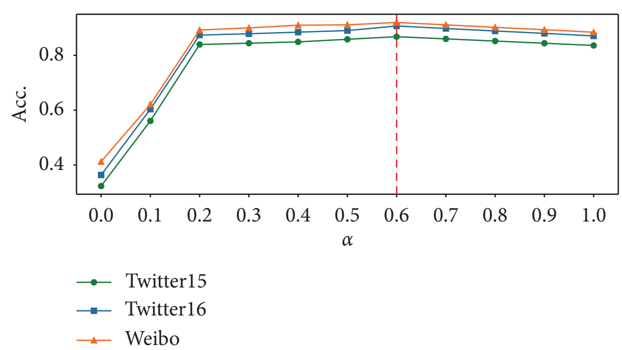
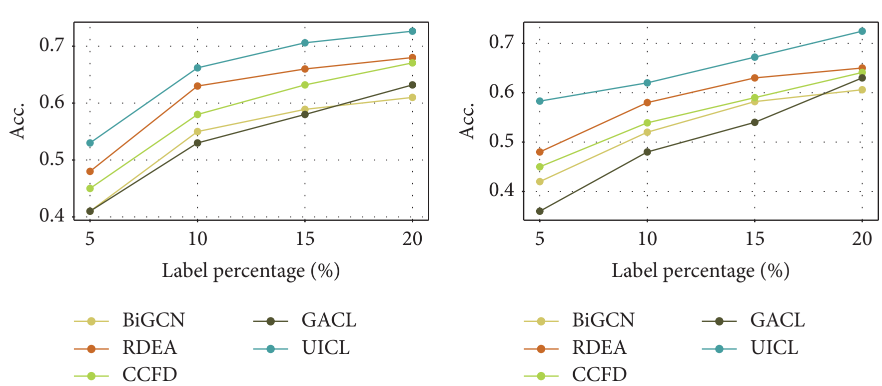
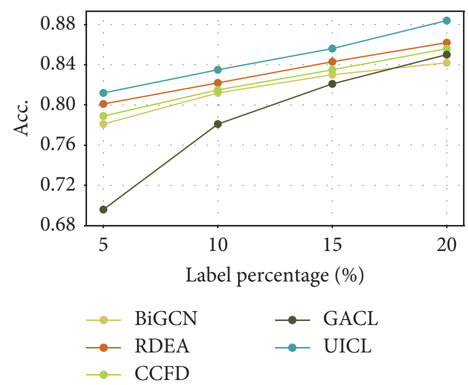


Figure 5: Results on different values of α.



(a) (b)

Figure 6: Continued.



(c)

Figure 6: Fine-tuning results with different label fractions. (a) Twitter15. (b) Twitter16. (c) Weibo.

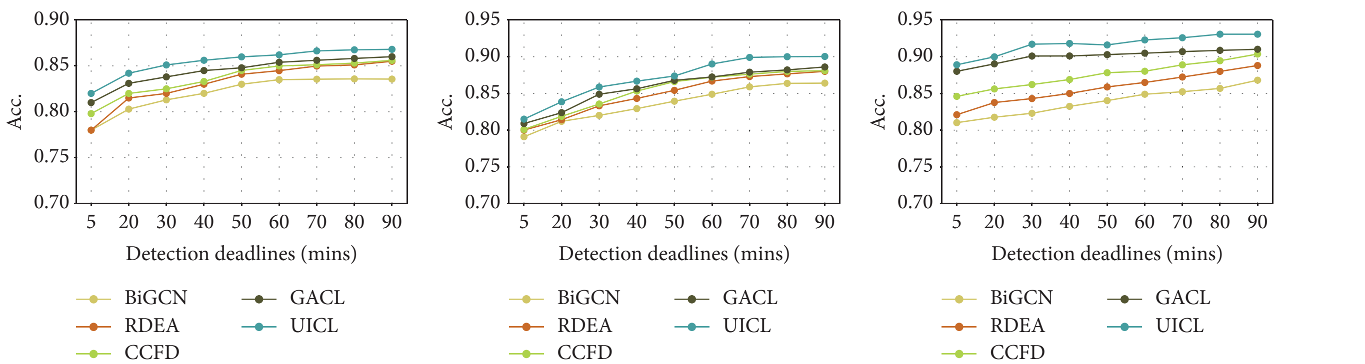


Figure 7: Results of rumor early detection on three datasets. (a) Twitter15. (b) Twitter16. (c) Weibo.

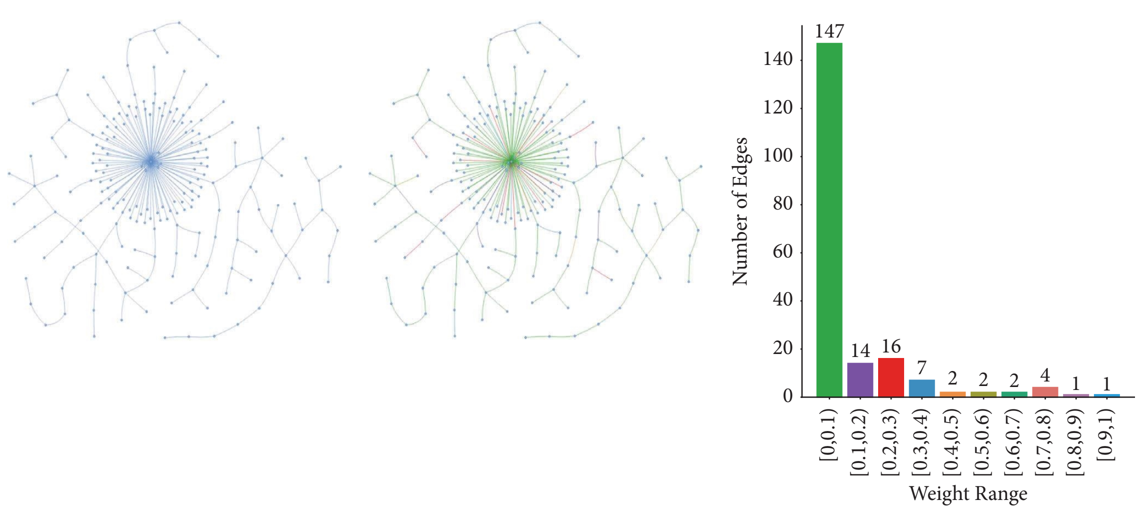


Figure 8: Visualization of the propagation structure and edge weight distribution after augmentation for a selected rumor event from the Twitter15 dataset. (a) Original propagation graph. (b) Augmented propagation graph. (c) Edge weight distribution.

these defined deadlines to construct the initial input propagation graph. In this experiment, we select Bi-GCN, CCFD, and GACL to compare with our model.

Figure 7 shows the performance comparison on early-stage rumor detection between UICL and selected baselines on three datasets. We observe a drop in performance for all

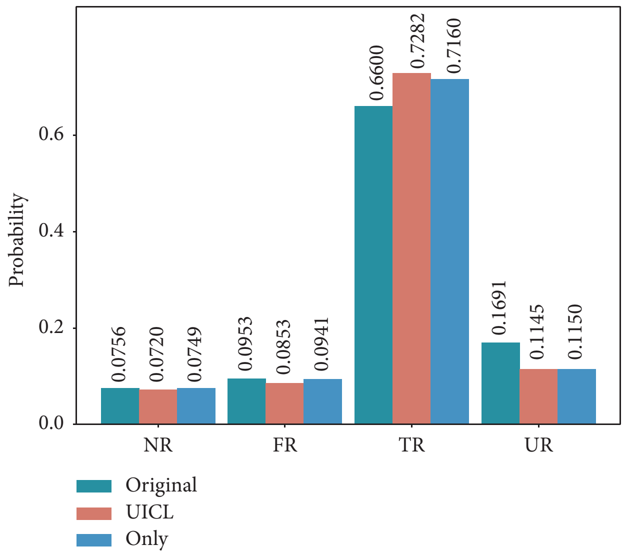


Figure 9: Rumor detection performance on different edge weight settings.

models when only a limited number of posts are tracked (within the 5-minute time window). Nevertheless, our UICL consistently outperforms other models, achieving accuracy of 83%, 81%, and 89% on Twitter15, Twitter16, and Weibo, respectively. The reason why UICL only needs a short time to identify rumors is that it can adaptively correct edge weights to learn more accurate structural features from the propagation graph. Furthermore, we can also observe that the performance of all methods improves over time, stemming from the increasing abundance of textual and structural information in the input propagation graph. However, the detection accuracy gradually becomes saturated due to the introduction of redundancy and noise.

5.5. Case Study. To demonstrate that the edge-inference augmentation module indeed enhances rumor detection accuracy, we conducted a case study by visualizing the reallocated edge weights after inference using edge-inference augmentation. Specifically, we randomly selected an event labeled as TR (ID 504131654810877952) from Twitter15 for further analysis. This event comprises 200 tweets, including the original post and its corresponding comments and retweets. Figure 8(a) plots its original propagation graph, which consists of 119 edges, each of them with an edge weight of 1, indicating that all edges are equally important for rumor detection. After edge-inference augmentation, the new propagation graph is generated with varying edge weights, as depicted in Figure 8(b), where different colors represent the various edge weight values.

Figure 8(c) depicts the specific distribution of edge weights. We find that a majority of the edges are denoted as less important, i.e., with an edge weight below 0.5. Thus, we conducted three experiments to verify the accuracy of the reallocated edges' importance by assessing the detection probability across different labels. Specifically, the settings for these three experiments are as follows: (1) Only: filters out edges with a weight less than 0.5, resets the weight of the remaining edges to one, and then uses them for detection. (2) UICL: utilizes all edges but with augmented edge weights. (3) Original: employs the original propagation graph, where each edge has the same edge weight of one for detection. The experimental results are illustrated in Figure 9. We observe that UICL achieves a higher probability value for TR, while yielding trivial probability distributions for other labels, demonstrating the effectiveness of the adjusted edge weights in rumor detection. Additionally, the results only further support this finding that using only the selected important edges, based on the adjusted weights from the edge-inference augmentation module, significantly improves model performance compared to Original.

6. Conclusion and Future Work

The critical role of graph structure in understanding and managing social network user behavior is fundamental. In this study, we introduce a novel deep learning framework, UICL, designed for effective rumor detection. UICL proposes an edge-wise augmentation strategy, comprising edge-inference and Dropedge augmentations. Edge-inference augmentation dynamically adjusts edge weights to reflect the inherent uncertainty in the propagation structures of real-world social networks. Specifically, it produces an augmented graph view by adaptively adjusting edge weights during the pretrained contrastive learning phase. Additionally, UICL utilizes DropEdge to generate another augmented graph view, enabling the capture of diverse graph structures in training samples. During the contrastive learning phase, UICL also introduces a tailored filtering mechanism for selecting more informative negative samples. Finally, we fine-tune the pretrained model with labeled data to enhance its predictive accuracy. Our experimental results demonstrate that UICL effectively reduces instability in social network propagation and outperforms other baseline methods in rumor detection. In future work, we aim to focus on advancing model interpretability within the rumor detection model.

Data Availability

The Twitter15/16 and Weibo data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China (U2333207) and by key R&D projects of Sichuan Provincial Science and Technology Plan (2023YFG0029).

References

[1] X. Chen, F. Zhou, F. Zhang, and M. Bonsangue, “Modeling microscopic and macroscopic information diffusion for rumor detection,” *International Journal of Intelligent Systems*, vol. 36, no. 10, pp. 5449–5471, 2021.

[2] W. Jing and H. Kang, “An effective ISDPR rumor propagation model on complex networks,” *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 11188–11213, 2022.

[3] W. Li, C. Guo, Y. Liu, X. Zhou, Q. Jin, and M. Xin, “Rumor source localization in social networks based on infection potential energy,” *Information Sciences*, vol. 634, pp. 172–188, 2023.

[4] C. Castillo, M. Mendoza, and B. Poblete, “Information credibility on twitter,” in *Proceedings of the 20th International Conference on World Wide Web*, pp. 675–684, ACM, Hyderabad, India, March 2011.

[5] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, “Prominent features of rumor propagation in online social media,” in *Proceedings of the 2013 IEEE 13th International Conference on Data Mining*, pp. 1103–1108, IEEE Computer Society, Dallas, TX, USA, December 2013.

[6] T. Chen, X. Li, H. Yin, and J. Zhang, “Call attention to rumors: deep attention based recurrent neural networks for early rumor detection,” in *Proceedings of theTrends and Applications in Knowledge Discovery and Data Mining - PAKDD 2018 Workshops*, pp. 40–52, Springer, Melbourne, Australia, June 2018.

[7] J. Ma, W. Gao, and P. Mitra, “Detecting rumors from microblogs with recurrent neural networks,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 3818–3824, IJCAI/AAAI Press, New York, NY, USA, July 2016.

[8] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[9] J. Ma, W. Gao, and K. Wong, “Rumor detection on twitter with tree-structured recursive neural networks,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pp. 1980–1989, Association for Computational Linguistics, Melbourne, Australia, July 2018.

[10] T. Bian, X. Xiao, T. Xu et al.,, “Rumor detection on social media with Bi-directional graph convolutional networks,” in *Proceedings of the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020*, pp. 549–556, AAAI Press, New York, NY, USA, February 2020.

[11] A. M. Luvembe, W. Li, S. Li, F. Liu, and G. Xu, “Dual emotion based fake news detection: a deep attention-weight update approach,” *Information Processing and Management*, vol. 60, no. 4, Article ID 103354, 2023.

[12] D. Chen, X. Chen, P. Lu, X. Wang, and X. Lan, “CNFRD: a few-shot rumor detection framework via capsule network for COVID-19,” *International Journal of Intelligent Systems*, vol. 2023, pp. 1–21, 2023.

[13] Z. He, C. Li, F. Zhou, and Y. Yang, “Rumor detection on social media with event augmentations,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2020–2024, ACM, Canada, July 2021.

[14] Q. Zhang, S. Zhang, J. Dong, J. Xiong, and X. Cheng, “Automatic detection of rumor on social network,” in *Proceedings of the Natural Language Processing and Chinese Computing - 4th CCF Conference, NLPCC 2015*, pp. 113–122, Springer, Nanchang, China, October 2015.

[15] S. E. Hihi and Y. Bengio, “Hierarchical recurrent neural networks for long-term dependencies,” in *Proceedings of the Advances in Neural Information Processing Systems 8, NIPS*, pp. 493–499, MIT Press, Denver, CO, USA, November 1995.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012*, pp. 1106–1114, Lake Tahoe, NV, USA, December 2012.

[17] Y. Liu and Y. B. Wu, “Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 354–361, AAAI Press, New Orleans, LA, USA, February 2018.

[18] A. Vaswani, N. Shazeer, and N. Parmar, “Attention is all you need,” in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

[19] L. M. S. Khoo, H. L. Chieu, Z. Qian, and J. Jiang, “Interpretable rumor detection in microblogs by attending to user interactions,” in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pp. 8783–8790, AAAI Press, New York, NY, USA, February 2020.

[20] T. Zou, Z. Qian, P. Li, and Q. Zhu, “Cross-modal adversarial contrastive learning for multi-modal rumor detection,” in *Proceedings of theIEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023*, pp. 1–5, IEEE, Rhodes Island, Greece, June 2023.

[21] M. Sun, X. Zhang, J. Ma, S. Xie, Y. Liu, and P. S. Yu, “Inconsistent matters: a knowledge-guided dual-consistency network for multi-modal rumor detection,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12736–12749, 2023.

[22] J. Zheng, X. Zhang, S. Guo, Q. Wang, W. Zang, and Y. Zhang, “MFAN: multi-modal feature-enhanced attention networks for rumor detection,” in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*, pp. 23–29, Vienna, Austria, July 2022.

[23] X. Chen, F. Zhou, G. Trajcevski, and M. Bonsangue, “Multi-view learning with distinguishable feature fusion for rumor detection,” *Knowledge-Based Systems*, vol. 240, p. 108085, 2022.

[24] L. Wei, D. Hu, W. Zhou, X. Wang, and S. Hu, “Modeling the uncertainty of information propagation for rumor detection: a neuro-fuzzy approach,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 2522–2533, 2024.

[25] H. Lin, J. Ma, M. Cheng, Z. Yang, L. Chen, and G. Chen, “Rumor detection on twitter with claim-guided hierarchical graph attention networks,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pp. 10035–10047, Association for Computational Linguistics, Punta Cana, Dominican Republic, November 2021.

[26] H. Lin, P. Yi, J. Ma et al.,, “Zero-shot rumor detection with propagation structure via prompt learning,” in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*, pp. 5213–5221, AAAI Press, Washington, DC, USA, February 2023.

[27] H. Ran and C. Jia, “Unsupervised cross-domain rumor detection with contrastive learning and cross-attention,” in *Proceedings of the Thirty-Seventh AAAI Conference on*

Artificial Intelligence, AAAI 2023, pp. 13510–13518, AAAI Press, Washington, DC, USA, February 2023.

[28] T. Sun, Z. Qian, S. Dong, P. Li, and Q. Zhu, "Rumor detection on social media with graph adversarial contrastive learning," in *Proceedings of the WWW'22: The ACM Web Conference 2022*, pp. 2789–2797, ACM, Lyon, France, April 2022.

[29] S. Li, W. Li, A. M. Luvembe, and W. Tong, "Graph contrastive learning with feature augmentation for rumor detection," *IEEE Transactions on Computational Social Systems*, pp. 1–10, 2024.

[30] H. Lin, J. Ma, L. Chen, Z. Yang, M. Cheng, and G. Chen, "Detect rumors in microblog posts for low-resource domains via adversarial contrastive learning," in *Proceedings of the-Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 2543–2556, Association for Computational Linguistics, Seattle, WA, USA, July 2022.

[31] H. Lin, J. Ma, R. Yang, Z. Yang, and M. Cheng, "Towards low-resource rumor detection: unified contrastive transfer with propagation structure," *Neurocomputing*, vol. 578, Article ID 127438, 2024.

[32] J. Qiu, Q. Chen, Y. Dong et al.,, "GCC: graph contrastive coding for graph neural network pre-training," in *Proceedings of the KDD'20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1150–1160, ACM, Virtual Event, CA, USA, August 2020.

[33] K. Hassani and A. H. K. Ahmadi, "Contrastive multi-view representation learning on graphs," in *Proceedings of the Machine Learning Research. Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, pp. 4116–4126, PMLR, Virtual Event, July 2020.

[34] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proceedings of the WWW'21: The Web Conference 2021*, pp. 2069–2080, ACM/IW3C2, Ljubljana, Slovenia, April 2021.

[35] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, Vancouver, Canada, December 2020.

[36] Y. Yang, Z. Wu, L. Wu et al.,, "Generative-contrastive graph learning for recommendation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023*, pp. 1117–1126, ACM, Taipei, Taiwan, July 2023.

[37] M. Yu, Z. Ding, J. Yu, W. Zhang, M. Yang, and M. Zhao, "Graph contrastive learning with adaptive augmentation for knowledge concept recommendation," in *Proceedings of the 26th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2023*, pp. 1281–1286, IEEE, Rio de Janeiro, Brazil, May 2023.

[38] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph contrastive representation learning," 2020, https://arxiv.org/abs/2006.04131.

[39] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, pp. 12121–12132, PMLR, Vienna, Austria, July 2021.

[40] D. Xu, W. Cheng, D. Luo, H. Chen, and X. Zhang, "InfoGCL: information-aware graph contrastive learning," in *Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pp. 30414–30425, NeurIPS, New Orleans, LA, USA, December 2021.

[41] H. Zhang, Q. Wu, J. Yan, D. Wipf, and P. S. Yu, "From canonical correlation analysis to self-supervised graph neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 76–89, 2021.

[42] H. Zhao, X. Yang, Z. Wang, E. Yang, and C. Deng, "Graph debiased contrastive learning with joint representation clustering," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, pp. 3434–3440, Montreal, Canada, August 2021.

[43] G. Chu, X. Wang, C. Shi, and X. Jiang, "CuCo: graph representation with curriculum contrastive learning," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, pp. 2300–2306, Montreal, Canada, August 2021.

[44] J. B. Grill, F. Strub, and F. Altché, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21271–21284, 2020.

[45] F. Zhao and D. Wang, "Multimodal graph meta contrastive learning," in *Proceedings of the CIKM'21: The 30th ACM International Conference on Information and Knowledge Management*, pp. 3657–3661, ACM, Queensland, Australia, November 2021.

[46] Y. Yin, Q. Wang, S. Huang, H. Xiong, and X. Zhang, "AutoGCL: Automated graph contrastive learning via learnable view generators," in *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pp. 8892–8900, AAAI Press, Vancouver, Canada, February 2022.

[47] S. Becker, "Mutual information maximization: models of cortical self-organization," *Network: Computation in Neural Systems*, vol. 7, no. 1, pp. 7–31, 1996.

[48] L. Wei, D. Hu, W. Zhou, Z. Yue, and S. Hu, "Towards propagation uncertainty: edge-enhanced bayesian graph convolutional networks for rumor detection," 2021, https://arxiv.org/abs/2107.11934.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[50] J. Ma, W. Gao, and K. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pp. 708–717, Association for Computational Linguistics, Vancouver, Canada, August 2017.

[51] J. Ma, Y. Liu, M. Liu, and M. Han, "Curriculum contrastive learning for fake news detection," in *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pp. 4309–4313, ACM, Atlanta, GA, USA, October 2022.

[52] Y. Zhu, Y. Xu, Q. Liu, and S. Wu, "An empirical study of graph contrastive learning," 2021, https://arxiv.org/abs/2109.01116.