Resolving Embedding-Ignoring Conflict in Graph Contrastive Learning-Based Rumor Detection

Jiachen Ma, Jing Dai, Wei Zhang(✉), and Yong Liu

School of Computer and Big Data, Heilongjiang University, Harbin, China
majiachen@hlju.edu.cn, 2221878@s.hlju.edu.cn, zhangwei_jsj@hlju.edu.cn,
liuyong123456@hlju.edu.cn

Abstract. Graph Contrastive Learning (GCL) has been widely used in the field of rumor detection due to its ability to self-supervise the extraction of low dimensional representations of propagation structure. However, existing methods focus on augmenting propagation structures and applying GCL at different levels for events, ignoring the conflict between InfoNCE-based contrastive loss and the message-passing mechanism in Graph Neural Networks (GNNs). This conflict prevents negative samples from effectively pulling away from each other, resulting in poor performance in rumor detection. To address this problem, we propose a novel GCL method for rumor detection called RERG. RERG detects gradient confusion to capture the conflicts and then ignores negative samples that cause conflicts, allowing GNNs to learn from the ignored negative samples adaptively. In addition, we propose a propagation structure augmentation method that considers edge importance to preserve the critical information of the propagation structure, thereby obtaining high-quality positive samples. Experimental results on public datasets prove that RERG achieves better results than other state-of-the-art models.

Keywords: Rumor detection · Graph contrastive learning · Graph neural networks · InfoNCE loss.

1 Introduction

With the significant success of GCL across various fields[9,8,7,10,19], many GCL models for rumor detection have been proposed recently. GCL-based methods for rumor detection mainly use the InfoNCE-based loss of GCL to pull the representations of positive pairs closer to each other and push the representations of negative pairs further apart. However, due to the conflict between the InfoNCE-based loss and the message-passing mechanism of GNNs, implicit conflicts that severely confuse the model arise when it learns from negative pairs.

As shown in Fig.1a and Fig.1c, we consider the example of a three-node propagation structure consisting of one source node v_1 and two related nodes v_2 and v_3, using a single-layer GNNs that includes each node's one-hop neighbors. Due to the message passing mechanism of GNNs, during forward propagation, v_1 aggregates features from itself and v_2, v_3. v_2 aggregates features from itself and

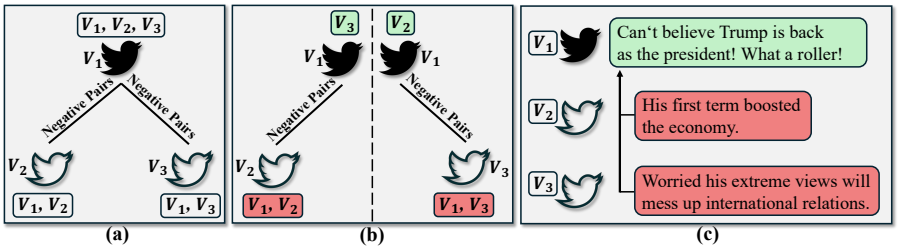

Fig. 1. Source node's relationship with related nodes in propagation structure.

v_1, while v_3 aggregates features from itself and v_1. However, during the process of GCL, both v_2 and v_3 form negative pairs with v_1. As shown in Fig.1b, for the negative pair formed by v_1 and v_2, the InfoNCE-based loss embeds features from v_3 to highlight the differences between the negative pairs while ignoring information from v_1 and v_2 to minimize their similarity. And for the negative pair formed by v_1 and v_3, the InfoNCE-based loss function embeds features from v_2 while ignoring information from v_1 and v_3. Thus, when the GNNs generates the representation of v_1, the InfoNCE-based loss requires the GNNs to both embed and ignore aggregated features from nodes v_2 and v_3.

The above conflict is referred to as the Embedding and Ignoring Conflict (EIC) and has been theoretically proven in detail by He et al.[3]. Since EIC severely affects the performance of GCL-based rumor detection methods, therefore, in this paper, we propose a novel GCL model called RERG to solve EIC in the field of rumor detection. As shown in Fig.2, first, for each propagation structure, we generate two augmented propagation structures by considering the importance of edges. Next, we design the negative sample capturer and negative sample controller. The negative sample capturer indirectly detects the confusion of the gradient by observing differences in the similarity of sample pairs across training iterations and thus captures the negative pairs that suffer from EIC. The negative sample controller then ignores these negative pairs, enabling the propagation structure encoder to adaptively learn from the ignored negative samples. The ignored nodes are updated in each iteration to ensure that all negative samples are utilized for training. Finally, We concatenate the augmented propagation structure as the final representation of the event and optimize both the cross-entropy loss and the contrastive loss. The main contributions of this paper can be summarized as follows:

- We propose a GCL method called RERG to eliminate EIC in the field of rumor detection.
- We propose a propagation structure augmentation method which consider the importance of edges to maintain the critical information.
- Experimental results on public datasets demonstrate the effectiveness of RERG.

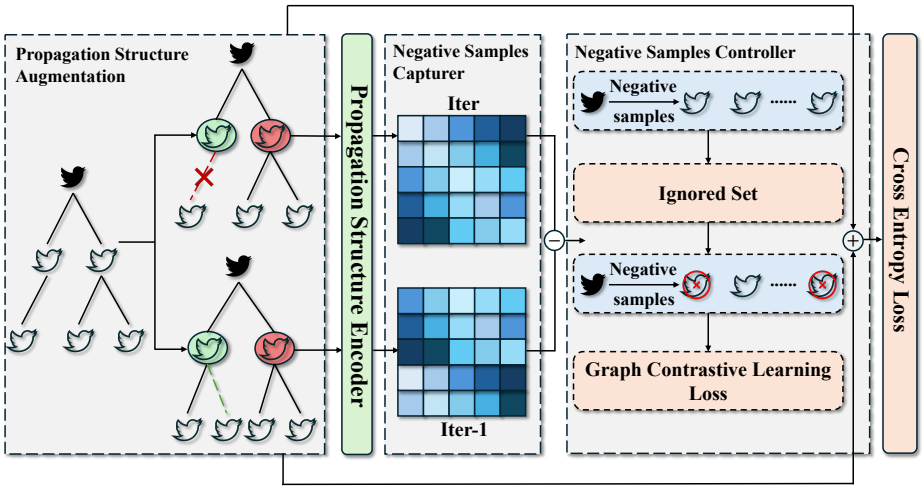

Fig. 2. The overall framework of RERG model.

2 Methodology

2.1 Problem Statement

$D = \{e_1, e_2, \ldots, e_n\}$ is defined as the rumor dataset, where e_i is the i-th event and n is the total number of events. $e_i = \{s_i, r_i^1, r_i^2, \ldots, r_i^{|v_i|-1}, G_i, y_i\}$, where s_i is the source post, r_i^j is the j-th related post which is related to s_i. $G_i = (V_i, E_i)$ is the propagation structure of the i-th event, where V_i and E_i are the nodes and edges of G_i respectively. If r_i^1 has a response to r_i^2, there will be a directed edge $r_i^1 \rightarrow r_i^2$. $y_i \in \{T, F, U, N\}$ (True Rumor, False Rumor, Unverified Rumor, and Non-rumor) is the ground-label of e_i. We define a feature matrix X_i and an adjacency matrix A_i from G_i. A_i embodies the forwarding relationships, X_i is initialized encoded by the pre-trained BERT model[15].

The goal of rumor detection is to train a classifier $f : \mathcal{D} \rightarrow \{T, F, U, N\}$ to predict the label of new events using labeled training data.

2.2 Propagation Structure Augmentation

We evaluate edge importance using edge centrality $w(r_i^a, r_i^b)$, determined by the centrality of the connected nodes. Node centrality $deg(\cdot)$ is measured by the node's degree. In directed propagation structures, the centrality of the tail node $deg(r_i^b)$ is used because the tail node significantly impacts the edge. During training, we apply edge augmentation operations (drop or add) based on the probability $P(r)$. When dropping an edge, we define $s^e_{r_i^a r_i^b} = \log w(r_i^a, r_i^b)$ to mitigate the effects of high-degree nodes. The probability of each edge being

dropped is defined as follows:

$$p_{r_i^a r_i^b} = \min\left(\frac{s^e_{\max} - s^e_{r_i^a r_i^b}}{s^e_{\max} - s^e_{\text{avg}}} \cdot p_e, p_r\right) \tag{1}$$

where $s^e_{\max}$ and s^e_{avg} are the maximum and average of $s^e_{r_i^a r_i^b}$, respectively. The parameter p_e controls the overall probability of dropping edges, and $p_r < 1$ is a cut-off probability that prevents the propagation structure from being overly corrupted. For the text content of the propagation structure nodes, we apply a dropout mask to create noisy text samples with minimal missing information. BERT model is utilized to encode the source post and its comments separately. We concatenate the source post and comments in the format "$[CLS]Source[SEP]Comment[SEP]$" to highlight the significance of the source post, using the final hidden state of the $[CLS]$ token as the corresponding node representation.

2.3 Propagation Structure Encoder

For each propagation structure G_i, we obtain two augmented propagation structures, $\tilde{G}'_i$ and $\tilde{G}''_i$, after augmentation. Correspondingly, the adjacency matrix A_i is converted into $\tilde{A}'_i$ and $\tilde{A}''_i$, and the feature matrix X_i is converted into $\tilde{X}'_i$ and $\tilde{X}''_i$. For one augmented propagation structure, $\tilde{G}'_i$, we use a two-layer GCN to capture the embeddings as follows:

$$H'_i = \sigma(\hat{A}'_i \sigma(\hat{A}'_i \tilde{X}'_i W_i^{(0)}) W_i^{(1)}), \tag{2}$$

where $W_i^{(k)}$ is the weight matrix and $\sigma()$ is the ReLU function. $\hat{A}'_i = \tilde{D}^{-\frac{1}{2}} \tilde{A}'_i \tilde{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix, where $\tilde{A}'_i = \tilde{A}'_i + I_N$, $\tilde{D}_{ii} = \sum_j \tilde{A}'_{ij}$ represents the degree of the i-th node.

2.4 Negative Samples Capturer

To effectively identify and handle EIC negative pairs in propagation structures, we design a negative samples capturer to identify negative pairs that cause gradient confusion. The negative samples capturer quantifies the gradient impact due to the introduction of particular negative samples as follows:

$$\nabla W_e = \frac{\partial L_{\text{InfoNCE}}}{\partial W_e} \tag{3}$$

where W_e denotes the parameters of propagation structure encoder. However, calculating ∇W_e directly for negative pairs presents mathematical difficulties. This is because the negative pairs that cause gradient confusion are not isolated, which requires evaluating their influence in combination with other negative pairs. Thus, during the training phase, we leverage the change in similarity caused by negative pairs to approximate their impact on the gradient as follows:

$$\Delta S_{ij} = S_{ij}^{iter} - S_{ij}^{iter-1} \tag{4}$$

where ΔS_{ij} is the gradient impact of the negative pair consisting of r_i and r_j. S_{ij}^{iter} is the cosine similarity between r_i and r_j at the current iteration. $iter$ denotes the iteration and $iter \geq 1$, and we set every N epochs as one iteration. For each iteration, we store the current S_{ij}^{iter}. If a negative pair is not involved in EIC, or if EIC minimally impacts the nodes in the negative pair, the similarity of the negative pairs will decrease rapidly, resulting in a smaller ΔS_{ij}. Conversely, if a negative pair is involved in EIC, the similarity will not decrease adequately and may even increase, leading to a larger ΔS_{ij}. Thus, we can indirectly detect whether the negative pair has been affected by EIC through the difference in similarity of negative samples between two iterations

2.5 Negative Samples Controller

We design a negative sample controller to create an ignore set by filtering out negative pairs with larger values in Eq(4), thereby ignoring negative pairs that would cause EIC during the training process as follows:

$$IgS = \{NP(i,j) \mid \Delta S_{ij} > \Delta S_r\}, \tag{5}$$

where $r\%$ is the ignore ratio, ΔS_r is the minimum value among the top r nodes when sorted in descending order. For negative pairs in the ignore set, the negative sample controller temporarily excludes them from self-supervised sampling. Initially, most negative pairs in the ignore set are significantly affected by EIC. However, after a period of training, some negative pairs participating in the training provide enough information, and their similarity no longer decreases, allowing them to enter the ignored set. At this point, the initially ignored negative pairs are reengaged in training, ensuring all negative pairs are used without causing EIC. Therefore, we ensure that all negative samples are involved in the training in this way.

2.6 Rumor Prediction

For each event e_i, we obtain the representations of two augmented propagation structures H_i' and H_i'' using Eq(2). These representations are concatenated to form the final representation of e_i as follows:

$$m_i = concat(H_i', H_i'') \tag{6}$$

Next, m_i is fed into full-connection layers and a softmax layer as follows:

$$\hat{y}_i = softmax(FC(m_i)) \tag{7}$$

where $\hat{y}_i$ is the predicted probability distribution, $FC(.)$ denotes full-connection layers.

2.7 Joint Training Procedure

We minimize the cross-entropy loss of the prediction $\hat{y}_i$ and ground truth y_i as follows:

$$\mathcal{L}_{cross} = -\sum_{i}^{|N|} y_i \log \hat{y}_i, \tag{8}$$

For the GCL loss, we amplify the gradients of negative samples using a parameter α to prevent the reduction of gradients caused by the ignoring of some nodes during training. Additionally, the exponential and logarithmic operations of the loss function further diminish these gradients. For positive samples, we introduce a parameter β, defined as follows:

$$\alpha = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} e^{\theta(z_i,z_j)/\tau}}{\sum_{i=1}^{n}\sum_{j=1,NP(i,j)\notin IgS}^{n} e^{\theta(z_i,z_j)/\tau}}. \tag{9}$$

$$\beta = \left(1 - \frac{1}{\alpha}\right) \cdot \frac{1}{N}\sum_{i=1}^{n} S_{ii'}. \tag{10}$$

where z_i or z_j denotes the embeddings of nodes in the augmented propagation structures. By using the parameters α and β, the gradient values remain relatively stable even when a large number of negative samples are ignored. This stability prevents the model from converging prematurely due to excessively large ignored partial values of r. Specifically, α is designed to dynamically adjust the influence of conflicting negative samples based on their similarity changes. This ensures that the model focuses on resolving gradient confusion caused by hard negative samples. Similarly, β controls the weight of the similarity preservation term to balance the retention of critical structural information and the overall contrastive objective. The GCL loss is defined as follows:

$$\mathcal{L}_{GCL} = -\log\Big(\frac{e^{S_{ii'}/\kappa}}{e^{S_{ii'}/\kappa} + \alpha\big(\sum_{k=1,NP(i,k)\notin IgS}\big(e^{S_{ik}/\kappa} + e^{S_{ik'}/\kappa}\big)\big)}\Big) - \beta. \tag{11}$$

where κ is the temperature parameter. The final loss consist of cross-entropy loss and GCL loss, which is defined as follows:

$$\mathcal{L} = \mathcal{L}_{cross} + \xi\mathcal{L}_{GCL} \tag{12}$$

During the training process, we dynamically adjusted the weights ξ to ensure that both losses were appropriately emphasized according to their current importance.

Table 1. Rumor detection results on Twitter15 and Twitter16 datasets.

Dataset	Twitter15					Twitter16				
Method	ACC.	N F1	F F1	T F1	U F1	ACC.	N F1	F F1	T F1	U F1
DTC	0.454	0.733	0.355	0.317	0.415	0.465	0.643	0.393	0.419	0.403
SVM-RBF	0.318	0.225	0.082	0.455	0.218	0.553	0.670	0.085	0.117	0.361
SVM-TK	0.750	0.804	0.698	0.765	0.733	0.732	0.740	0.709	0.836	0.686
RvNN	0.723	0.682	0.758	0.821	0.654	0.737	0.662	0.743	0.835	0.708
BiGCN	0.836	0.791	0.842	0.887	0.801	0.864	0.788	0.859	0.932	0.864
DDGCN	0.835	0.840	0.850	0.856	0.791	0.893	0.807	0.931	0.946	0.871
RDEA	0.855	0.831	0.857	0.903	0.816	0.880	0.823	0.878	0.937	0.875
CCFD	0.856	0.848	0.861	0.893	0.816	0.886	0.819	0.884	0.954	**0.892**
GACL	0.846	0.859	0.845	0.866	0.812	0.891	0.802	0.929	0.945	0.872
PFNC	0.874	0.848	**0.952**	0.888	0.817	0.911	0.868	0.924	0.956	0.850
DCRD	0.855	0.827	0.861	0.907	0.817	0.888	0.829	0.886	0.940	0.888
RERG-EDGE	0.874	0.860	0.871	0.941	0.844	0.912	0.903	0.923	0.947	0.844
RERG-EIC	0.867	0.835	0.872	0.943	0.812	0.904	0.880	0.892	0.951	0.842
RERG-ALL	0.865	0.817	0.853	0.937	0.800	0.898	0.872	0.895	0.949	0.825
RERG	**0.883**	**0.862**	0.885	**0.959**	**0.850**	**0.921**	**0.908**	**0.950**	**0.970**	0.886

3 Experiment

3.1 Datasets and Baselines

RERG is evaluated on two real-world datasets: Twitter15 [13] and Twitter16 [13]. Both Twitter15 and Twitter16 contain four categories: Non-rumor (N), False Rumor (F), True Rumor (T), and Unverified Rumor (U), which are used for quaternary classification. The datasets are randomly split for 5-fold cross-validation. Accuracy (Acc.) and F1-measure (F1) are used as evaluation metrics. We compare RERG with the following state-of-the-art baselines. **Machine learning methods:** DTC[2], SVM-RBF[18] and SVM-TK[13]. **GNN-based methods:** RvNN[14], BiGCN[1] and DDGCN[16]. **GCL-based methods:** RDEA[4], CCFD[12], GACL[17], PFNC[11] and DCRD[6].

3.2 Experiment Settings

RERG is implemented using PyTorch. The parameters p_e, p_r, κ, r, ξ, N and $P(r)$ are set to 0.1, 0.7, 0.5, 5, 0.001, 20 and (0.5, 0.5), respectively. The learning rate is initialized at 5×10^{-4} and decreases gradually during training according to a decay rate of 1×10^{-4}. The parameters of RERG are updated using gradient descent and optimized with the Adam algorithm [5]. The source code of RERG is available at `https://github.com/mjccn/dasfaa2025rerg`.

3.3 Results and Analysis

Table 1 presents the performance of all comparison methods on the Twitter15 and Twitter16 datasets. Bold text indicates the best performance, while underlined text denotes the second-best performance. The experimental results highlight the following key points:

- Deep learning significantly outperforms traditional machine learning, demonstrating its superior ability to automatically extract complex features from large datasets and identify subtle patterns in rumors.
- DDGCN outperforms Bi-GCN and RvNN, demonstrating its superior ability to model dynamic structures and capture propagation structures complexity.
- CCFD outperforms RDEA and DCRD across multiple metrics, indicating that progressively introducing challenging negative samples and using tailored event augmentation enhances the performance and robustness of GCL-based methods.
- Both GACL and PFNC achieve superior performance by enhancing robustness and feature representation. GACL uses adversarial training and GCL to simulate adversarial scenarios, while PFNC employs node-level GCL and data augmentation to improve propagation structure representation.
- RERG achieves the best performance, demonstrating the effectiveness of our proposed method in addressing the EIC problem for rumor detection by indirectly capturing negative pairs through node similarity, thereby reducing gradient confusion.

3.4 Ablation Study

To verify the effectiveness of the different modules of RERG, we perform a series of ablation studies:

- **RERG-EDGE** augments the propagation structure by randomly dropping edges without considering their importance.
- **RERG-EIC** directly optimizes the InfoNCE-based loss without considering the EIC problem in GCL methods.
- **RERG-ALL** augments propagation structures by randomly removing edges, without considering edge importance or the EIC problem in GCL, and then optimizes the InfoNCE-based loss.

The experimental results are shown in Table 1, it can be observed:

- The performance of RERG-EDGE decreased by 0.9% compared to RERG on both Twitter15 and Twitter16 datasets. Disregarding edge importance in propagation structure augmentation may cause loss of critical information, negatively impacting GCL effectiveness.
- RERG-EIC showed a performance decrease of 1.6% on Twitter15 dataset and 1.7% on Twitter16 dataset compared to RERG. This suggests that using GCL for rumor detection without addressing the EIC issue can create implicit conflicts that confuse the model.
- The performance of RERG-ALL, which lacks all key modules, was significantly lower, further underscoring the effectiveness of our proposed method.

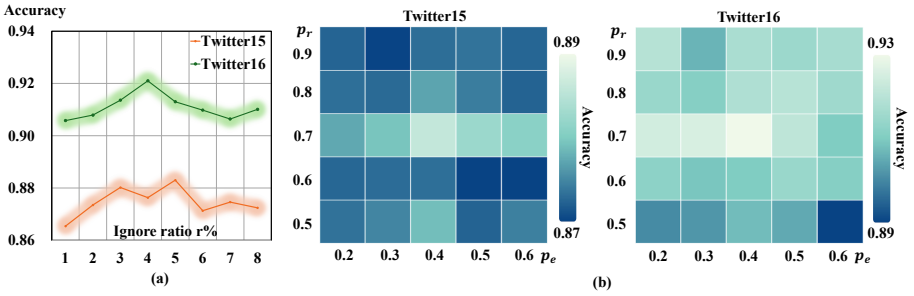

Fig. 3. (a) Analysis of parameter ignore radio r. (b) Analysis of parameters overall probability of dropping edges p_e and cut-off probability p_r.

3.5 Parameters Analysis

Fig.3(a) shows the accuracy of RERG on Twitter15 and Twitter16 datasets at various ignore ratios r. On Twitter15 dataset, accuracy increases as r rises from 1% to 5%, but declines after further increases. On Twitter16 dataset, accuracy improves with r from 1% to 4%, then decreases with higher r. These results suggest that ignoring some negative samples improves accuracy, but too high r reduces negative pairs for training, degrading performance.

Fig.3(b) shows the effectiveness of the overall probability p_e and cut-off probability p_r in Eq(1). The impact of edge importance is evaluated by varying p_e (0.2–0.6) and p_r (0.5–0.9). RERG performance improves as p_e and p_r increase, peaking at $p_e = 0.4$ and $p_r = 0.7$ on both datasets. Beyond these values, performance declines. Initially, unreliable relationships weaken model robustness, but increasing p_e and p_r reduces them, improving performance. However, excessively high values of p_e and p_r remove key edges, degrading performance.

4 Conclusion

In this paper, we propose a GCL-based model RERG that addresses the EIC problem in rumor detection by considering edge importance to generate augmented propagation structures and adaptively ignoring EIC-affected negative pairs. Experimental results show that RERG effectively improves rumor detection by mitigating EIC.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 6247074060), the Natural Science Foundation of Heilongjiang Province in China (No. PL2024F029), the Fundamental Research Funds for Province-owned Higher Education Institutions in Heilongjiang Province (2021-KYYWF-0043, 2023-KYYWF-1463, 2024-KYYWF-0115) and the Doctoral Postdoctoral Funding Project in Heilongjiang Province.

References

1. Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., Huang, J.: Rumor detection on social media with bi-directional graph convolutional networks. In: AAAI. vol. 34, pp. 549–556 (2020)
2. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: WWW. pp. 675–684 (2011)
3. He, D., Zhao, J., Huo, C., Huang, Y., Huang, Y., Feng, Z.: A new mechanism for eliminating implicit conflict in graph contrastive learning. In: AAAI. vol. 38, pp. 12340–12348 (2024)
4. He, Z., Li, C., Zhou, F., Yang, Y.: Rumor detection on social media with event augmentations. In: SIGIR. pp. 2020–2024 (2021)
5. Kingma, D.P.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Liu, H., Xue, Y., Yu, X.: Disentangled graph representation with contrastive learning for rumor detection. In: ICASSP. pp. 6470–6474 (2024)
7. Liu, M., Liang, K., Hu, D., Yu, H., Liu, Y., Meng, L., Tu, W., Zhou, S., Liu, X.: Tmac: Temporal multi-modal graph learning for acoustic event classification. In: ACM MM. pp. 3365–3374 (2023)
8. Liu, M., Liang, K., Zhao, Y., Tu, W., Zhou, S., Gan, X., Liu, X., Kunlun, H.: Self-supervised temporal graph learning with temporal and structural intensity alignment. TNNLS (2024)
9. Liu, M., Liu, Y., Liang, K., Tu, W., Wang, S., Zhou, S., Liu, X.: Deep temporal graph clustering. In: ICLR (2024)
10. Ma, J., Dai, J., Liu, Y., Han, M., Ai, C.: Contrastive learning for rumor detection via fitting beta mixture model. In: CIKM. pp. 4160–4164 (2023)
11. Ma, J., Liu, Y., Han, M., Hu, C., Ju, Z.: Propagation structure fusion for rumor detection based on node-level contrastive learning. TNNLS **35**(12), 18649–18660 (2024)
12. Ma, J., Liu, Y., Liu, M., Han, M.: Curriculum contrastive learning for fake news detection. In: CIKM. pp. 4309–4313 (2022)
13. Ma, J., Gao, W., Wong, K.: Detect rumors in microblog posts using propagation structure via kernel learning. In: ACL. pp. 708–717 (2017)
14. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J.: Rumor detection on twitter with tree-structured recursive neural networks. In: ACL. pp. 1980–1989 (2018)
15. Nguyen, D.Q., Vu, T., Tuan Nguyen, A.: Bertweet: A pre-trained language model for english tweets. In: EMNLP. pp. 9–14 (2020)
16. Sun, M., Zhang, X., Zheng, J., Ma, G.: Ddgcn: Dual dynamic graph convolutional networks for rumor detection on social media. In: AAAI. vol. 36, pp. 4611–4619 (2022)
17. Sun, T., Qian, Z., Dong, S., Li, P., Zhu, Q.: Rumor detection on social media with graph adversarial contrastive learning. In: WWW. pp. 2789–2797 (2022)
18. Yang, F., Liu, Y., Yu, X., Yang, M.: Automatic detection of rumor on sina weibo. In: KDD. pp. 1–7 (2012)
19. Zhang, L., Ma, J., Fu, B., Lin, F., Sun, Y., Wang, F.: Improved central attention network-based tensor rx for hyperspectral anomaly detection. Remote Sensing **14**(22), 5865 (2022)