Joint Stance and Rumor Detection in Hierarchical Heterogeneous Graph

Chen Li, Peng Hao, Jianxin Li, Lichao Sun, Lingjuan Lyu, Lihong Wang, and Philip S. Yu, *Fellow, IEEE*, Lifang He

***Abstract*—Recently, large volumes of false or unverified information (e.g., fake news and rumors) appear frequently in emerging social media, which are often discussed on a large scale and widely disseminated, causing bad consequences. Many studies on rumor detection indicate that the stance distribution of posts is closely related to the rumor veracity. However, these two tasks are generally considered separately or just using a shared encoder/layer via multi-task learning, without exploring the more profound correlation between them. In particular, the performance of existing methods relies heavily on the quality of hand-crafted features and the quantity of labeled data, which is not conducive to early rumor detection and few-shot detection. In this paper, we construct a hierarchical heterogeneous graph by associating posts containing the same high-frequency words to facilitate the feature cross-topic propagation, and jointly formulate stance and rumor detection as multi-stage classification tasks. To realize the updating of node embeddings jointly driven by stance and rumor detection, we propose a Multi-GNN framework, which can more flexibly capture the attribute and structure information of the context. Experiments on real datasets collected from Twitter and Reddit show that our method outperforms state-of-the-art by a large margin on both stance and rumor detection. And the experimental results also show that our method has better interpretability and requires less labeled data.**

***Index Terms*—Rumor Detection, Stance Detection, Hierarchical Heterogeneous Graph, Adaptive Graph Attention, Graph Pooling.**

I. Introduction

EVERY day, billions of people would use social media to browse news, share opinions, and interact with others in real-time. Social media has notably improved the diffusion speed and range of information [1]. However, some abnormal communication in social media can directly lead to harmful effects, interfering with people's timely and accurate access to information [2]. In particular, the content of information on social media often lacks necessary management, and thus large amounts of fake or unverified information will be released and mixed with the real ones. The popularity of social media makes it convenient for information acquisition, but virtually intensifies the influence of rumors on the Internet and even the real world. Hence, detecting rumors circulating in social media as early as possible is a very important task [3].

The sociological definition of the rumor is "unverified and instrumentally relevant information statements in circulation" [4], and the task of rumor detection aims to determine the veracity of given information. By treating rumor detection as a supervised text classification task, most traditional methods [5], [6] collect and encode vast hand-crafted features from post content, user profiles, and propagation patterns to train an effective classifier. Recently, to automatically learn the feature representations in a data-driven way, several deep neural networks based methods or tree-kernels based methods have been proposed [7]–[10]. In particular, some studies focus on taking the attitude of the post as an essential signal to determine the veracity of rumors [11]–[15]. It involves another research topic that is closely related to rumor detection: *stance detection*, which aims to determine the attitude (i.e., SDQC) of a given text toward a specific target. Similar to the traditional text classification task, several semantic and statistical feature-based methods have been applied to categorize the stance of the given text. Among them, two types of methods were widely used for feature construction: one is the feature engineering and statistical model [6], [16], and the other is deep neural network-based model [17]–[19].

However, there are still some limitations in previous meth-

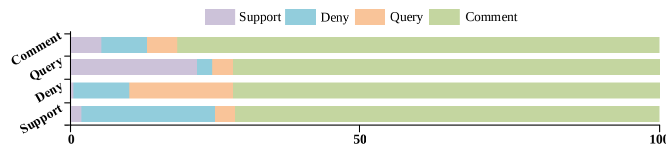


Fig. 1. The stance distribution in RumourEval2019. Different colored bars represent the proportion of its contextual stances of a specific stance, including support, deny, query and comment (SDQC).

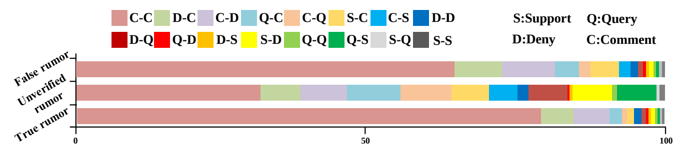


Fig. 2. The distribution of user interactions under different veracity of rumors. 'SDQC' are the acronyms for stances. For instance, 'Q-S' is one kind of user interaction, which indicates a post with 'Query' stance is commented by a post with 'Support' stance.

Chen Li, Hao Peng and Jianxin Li are with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China. (e-mail: lichen@act.buaa.edu.cn, penghao@act.buaa.edu.cn, lijx@act.buaa.edu.cn).

Lichao Sun and Lifang He are with the Lehigh University, Bethlehem, USA. (e-mail: James.lichao.sun@gmail.com, lih319@lehigh.edu).

Lingjuan Lyu is with the National University of Singapore, Singapore, Singapore. (e-mail: lingjuanlvsmile@gmail.com).

Lihong Wang is with the National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China. (e-mail: wlh@isc.org.cn).

Philip S. Yu is with the University of Illinois at Chicago, Chicago, USA. (e-mail: psy@uic.edu).

This paper is a submittion for special issue on Deep Learning for Anomaly Detection.

Manuscript received Nov. 28, 2020. (Corresponding author: Jianxin Li.)

U1: **We understand there are two gunmen and up to a dozen hostages inside the ... [support]**
├── **U2:** **@U1 not ISIS flags [deny]**
├── **U3:** **@U1 sorry – how do you know it is an ISIS flag? Can you actually ... [query]**
│   └── **U4:** **@U3 no she cannot cos it is actually not [deny]**
├── **U5:** **@U1 More on situation at Martin Place in Sydney, AU-LINK- [comment]**
└── **U6:** **@U1 Have you actually confirmed its an ISIS flag or you talking shit [query]**

Fig. 3. An example of rumor and several user interactions.

ods that need to be further addressed in detecting stance and rumor simultaneously. Existing stance detection methods [16], [20] consider either the text and attribute features of the post itself or the sequence relationship between posts in isolation but ignore the valuable local neighbor stance distribution information. As illustrated in Figure 1, except for a lot of 'Comment', the neighbor distribution of each stance has its unique characteristics (e.g., the proportion of 'Support' in the neighbor of 'Query' is significantly larger than the other two stances). This distribution depends on the inherent social network user interaction (i.e., a round of conversation, as shown in Figure 3) habits and has stability. Recently, some methods have attempted to encode structural information into models using graph convolution [21], [22] or graph attention [23], [24], but they are still limited to a single propagation tree and obtain neighborhood information uniformly (i.e., first/second-order neighbors). Besides, some researches [3], [25] on rumor detection pay more attention to sequentially encode the propagation tree but ignore the relationship between rumor veracity and the global distribution of user interaction types in the information diffusion [24], [26]. Figure 2 indicates that unverified rumors often lead to intense discussions (i.e., with a more balanced distribution of interaction types). In contrast, the proportion of comment class interactions in verified rumors (true/false) is more than three-quarters. Meanwhile, how to relate stance and rumor detection tasks, how to reduce the cost of labeled data, and early rumor detection are also issues that need to be considered in practical applications [3].

Inspired by the success of Graph Neural Networks (GNNs) [27]–[32], we jointly conduct stance and rumor detection in a uniform hierarchical heterogeneous graph, which does not fall in the category of classic joint learning with a weight-sharing setting. Unlike the recent approaches based on multi-task learning that jointly detect rumor and stance by the shared layers/features/encoder [9], [11], [13], we treat them as two phases of a complete task to share training gradients. Specifically, instead of sharing components to achieve multi-task learning, we update the node embeddings in the hierarchical heterogeneous graph driven by two tasks simultaneously. First, to correlate posts that may have the same stance across different rumors and then diffuse label information, we regard high-value words as a bridge to map the rumor corpus into a connected heterogeneous graph, where the nodes were single posts and words, and the edges were divided into three categories: *word-word*, *word-post*, and *post-post*. Second, to satisfy the basic assumptions of the label propagation and capture the characteristics of neighbor distribution, we use edge weights to adjust the proportion of information getting from neighbors and employ adaptive graph attention networks to divide the neighborhood flexibly and learn the representation of posts. Third, through coarsening a single propagation tree (including both node and structural features) into a node using a learnable structure sensitive graph pooling layer, the original graph is turned into a new heterogeneous graph with rumors and words as nodes. Intuitively, both the stance and rumor detection tasks are transformed into node classification tasks on the graph. In terms of the stance and rumor detection task datasets, our method significantly outperforms the state-of-the-art baselines in both tasks. Meanwhile, we also report the experimental results of early detection and visualization, which indicates that our model is more adapted to real application needs.

Compared to traditional methods, our method has several advantages:

- We are the first to integrate multiple rumor propagation trees into a unified hierarchical heterogeneous graph, and propose a novel joint stance and rumor detection method based on GNNs, which captures not only cross-topic characteristics and structural information more flexibly.
- With the help of the message passing network, the proposed framework has been verified that it can significantly reduce the amount of labeled data and facilitate the early detection of rumors.
- The experimental results demonstrate that our model outperforms or achieves state-of-the-art results on both stance and rumor detection.

The rest of this paper is organized as follows: Section II reviews traditional stance and rumor detection methods and the graph neural networks. Section III details the composition of the whole framework and each of the component proposed in this paper and conducts a theoretical analysis. Section IV introduces the experimental details and analyzes the results of typical downstream tasks. At length, Section V concludes this paper.

II. Related Work

Since this paper covers the contents of three research fields, we will introduce the related work from three perspectives.

A. Rumor Detection

The early works modeled rumor detection as a general supervised classification task, which tried to differentiate social information by manually defining and identifying the features of rumors. Among them, the work of Castillo et al. [33] extracted and combined features from multiple sources, and some subsequent methods introduced a wide range of social features [34]–[36]. Besides, some works [37] focused on modeling temporal features of rumor diffusion, and mixed them with traditional features as the basis for classification.

Different from the feature engineering based methods, many neural network based methods were proposed to learn features from diverse information automatically. Ma et al. [3] and Rath et al. [25] utilized recurrent neural network to learn the rumor representations and diffusion from post content and user interactions. Ma et al. [7] proposed a context-sensitive propagation tree kernel to capture high-order patterns for differentiating various types of rumors. Besides, tree-structured based recursive model [8], [12], graph kernel [38], hierarchical attention mechanism [10], multi-task learning [9], [11], [13], [30] and generative adversarial learning [39], [40] are also used to enhance the ability of modeling rumor features. Recently, some methods [21]–[23], [41] begin to consider the information contained in the stance distribution in the process of propagation, but are still limited to fixed depth neighbors, lack of flexibility. Furthermore, these methods only process rumors in isolation without considering shared features between topics.

However, existing methods generally ignore the information contained in the stance distribution in the process of propagation, and only process rumors in isolation without considering shared features between topics. To the best of our knowledge, our work is the first to jointly conduct stance and rumor detection on the hierarchical heterogeneous graph structure, and uses graph neural networks to learn better representations of posts and rumors.

B. Stance Detection

Stance detection has attracted increasing attention from the research community due to its importance for several downstream tasks. Among the tasks related to rumor detection, stance detection is generally defined as understanding the attitude of the post to target rumor. Most previous works relied on a diverse form of content-based features and supervised learning techniques. Recently, Lukasik et al. [42] and Lukasik et al. [43] noticed the association of stances between connected tweets, and respectively used Gaussian Process and Hawkes Process to model temporal sequence features of rumors. Besides, Ma et al. [9] further explored the relationship between the different propagation paths of the same rumor, and utilized weight sharing to extract the common and task-invariant features. Similarly, some deep neural network based methods have been proposed to achieve stance detection. The work of Kochkina et al. [17] utilized bidirectional long short term memory to encode the target tweet, and then used a recurrent neural network to classify the tweet sequence in turn. Instead of Recurrent Neural Network (RNN), Chen et al. [44] employed convolutional neural network to obtain the semantic embedding of the target tweet, and equipped a softmax classifier to determine the stance. Cheng et al. [30] defined four tasks to realize more task-driven feature extraction. Recently, some work [6], [16], [18] introduced pre-trained language models or extra corpus and combined semantic information with a wide range of hand-crafted features to achieve better detection results. However, the existing methods generally ignore the relationship between stance distribution and rumor veracity and do not explore the commonality of stance distribution, semantic information, and structural features among different topics.

To the best of our knowledge, our work is the first to conduct and achieve stance and rumor detection on hierarchical heterogeneous graph structure through a graph neural network based architecture, which is able to correlate different rumors and efficiently capture user interaction characteristics. In particular, different from multi-task learning using shared layers/features/encoder, our method proposes a tandem architecture that only shares the gradients of different tasks, simplifying the model structure on the premise of keeping task-driven.

C. Graph Neural Network

The research of graph neural networks has received much more attention, and much current work has extended the mature mechanism to arbitrarily structured graphs.

By defining the text as a graph in different forms, recent work begins to apply graph neural networks for text classification. Kipf et al. [27] achieved the state-of-the-art effect in the semi-supervised classification of citation network by first-order approximation of spectral convolutional networks. Velickovic et al. [28] introduced multi-head graph attention to learn more interpretable representations. The work of Peng et al. [45] transformed text into word graph, and then employed graph Convolution Neural Network (CNN) for sequence convolution. A novel Long-Short Term Memory (LSTM) structure for encoding text was proposed by [46], which changed the input mode of incremental reading. By constructing a heterogeneous graph of documents and words, Yao et al. [29] transformed the task of text classification into node classification on the graph structure. In addition, due to the density of the graph structure, recent methods began to divide the node neighborhood more carefully, in order to learn more robust and efficient features [47]–[49]. Specifically, these methods fully mine the structural information, adaptively divide the neighborhood of the target node, and more flexibly capture the neighborhood characteristics. In this paper, we introduce the graph neural network to help model fully capture the cross-topic information in the heterogeneous network, and then obtain richer semantic features.

As a new research hotspot in graph classification, pooling in convolution computation is used to realize down-sampling on feature graphs. Due to the limitation of non-European graph data structure, the pooling of the graph is different from the image pooling of the given step size and type of pooling. The hard rule is one of the simplest graph pooling

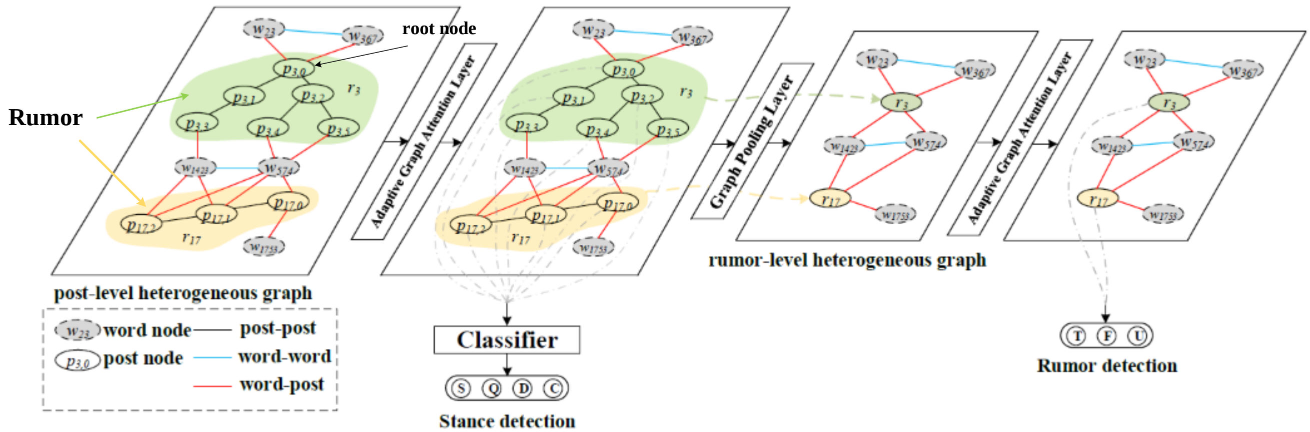


Fig. 4. The overall architecture of the proposed model. The whole process in the figure from left to right is to construct a hierarchical heterogeneous graph. The representation learning of post/rumor nodes is achieved through stacking different GNNs. The post-level heterogeneous graph is composed of post propagation tree and high-frequency words, in which there are two kinds of nodes (i.e., *word* and *post*) and three kinds of different edges. The rumor-level heterogeneous graph is a new graph obtained by the downsampling of the post-level heterogeneous graph through the graph pooling operation. There are two kinds of nodes (i.e., *word* and *rumor*) in the rumor-level heterogeneous graph, in which the edges are inherited from the post-level heterogeneous graph. Since we construct the labeled and unlabeled data in a unified hierarchical graph and perform representation learning, it is actually a transductive learning model.

operations that achieve node merging by pre-pooling nodes on a given graph structure. In order to make the process of node merging more flexible, some trainable rule models are proposed. Graph coarsening is a mainstream graph pooling method. Diffpool [50] has achieved the effect of pooling by soft clustering the nodes and then generating supernodes. Similarly, there are some methods that design different node selection strategies to achieve graph pooling. The work of Lee et al. [51] proposed a scoring sorting method for nodes in the class cluster to realize node selection and finally complete graph pooling.

In this paper, we introduce graph pooling to realize downsampling for a post propagation tree to generate a rumor node. Since the post propagation tree is a predefined graph structure, this operation is similar to a hard rule operation. However, in order to incorporate the impact of the stance of the posts in this pooling operation, we implemented it using constraints with trainable weights.

III. Methodology

In this section, we first give the problem definition of stance and rumor detection, and then describe the method of constructing rumor corpus into heterogeneous graph. Then, we propose Multi-GNN framework to realize the above two tasks at different stages.

A. Problem Definition and Notations

The definition of stance and rumor detection in this work partially follows the previous work [52]. The data provided by the rumor dataset is a collection of tree-structured conversations formed of posts replying to the originating rumorous post (as shown in Figure 3), i.e., rumors: $R = \{r_1, r_2, \cdots, r_N\}$, where each rumor r_i consists of a root post and its propagation tree structure, $N = |R|$. Each post $p_{i,j} \in r_i, j = 0, 1, \cdots, n-1$ presents its own attitude with regard to the rumor (i.e., the root post $p_{i,0}$), where n is the number of posts in r_i.

- **Stance Detection:** In this work, stance detection aims to determine the user's attitude with regard to the veracity of a given rumor r_i. Specifically, we need to label the type of interaction between a given statement (i.e., $p_{i,0}$) and a reply post $p_{i,j}$ in the rumor r_i, that is $f_{sd} : p_{i,j} \to y_{i,j}$, where $y_{i,j}$ is the label that takes one of the Supporting, Denying, Querying and Commenting (SDQC). As shown in Figure 3, each label represents the opinion contained in the post, however, the post with the label 'Comment' does not contain a clear attitude. In particular, as illustrated in Figure 1, the vast majority of posts do not contain a specific attitude in the real world.
- **Rumor Detection:** We formulate this task as a propagation tree classification problem, which learns a supervised classifier f_{rd} to differentiate a given rumor r_i. That is $f_{rd} : r_i \to y_i$, where y_i is one of the three candidate labels (True, False or Unverified).

B. Hierarchical Heterogeneous Graph Construction

As shown in Figure 4, we build a hierarchical heterogeneous graph which contains high frequency words as bridge for associating different rumors and share semantic information between different rumors. Specifically, the nodes in the post-level heterogeneous graph can be divided into two categories:

- *post node*, $p_{i,j}$ is the opinion from a certain user in specific rumor r_i. Among them, each post has at least one linked post, and every two linked posts form a complete user interaction process.
- *word node*, w_k is a high-frequency word selected from corpus. Each high-frequency word is an independent word, excluding words such as stop words without actual semantic information.

According to the different types of linked nodes, there are three different types of edges in the hierarchical heterogeneous graph. There are some differences in the specific meanings of these three kinds of edges, as shown below:

- *word-word*
  The purpose of inserting the high-frequency word nodes is to establish links between posts on different topics/propagation trees and to realize the flexible propagation of features. This operation is based on the common assumption in the graph neural network (i.e., the closer nodes are more likely to share similar features), so we need to calculate the weight of such edge by counting the co-occurrence frequency between high-frequency words in the corpus. Then, the relationship between word nodes w_i and w_j can be determined by the co-occurrence frequency in corpus, whose weight is calculated through Pointwise Mutual Information (PMI):
  $$\mathrm{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}. \tag{1}$$
  To simplify the calculation [29], we scan the whole corpus using sliding windows, and use N_{w_i}/N_{window} approximates $p(w_i)$, where N_{w_i} and N_{window} respectively denote the number of windows that w_i appears and all windows. Then, the Eq. 1 can be rewritten as:
  $$\mathrm{PMI}(w_i, w_j) = \log \frac{N_{w_i,w_j} \times N_{window}}{N_{w_i} \times N_{w_j}}, \tag{2}$$
  where N_{w_i,w_j} is the number of windows that w_i and w_j appear simultaneously.
- *word-post*
  Such edges are directly determined by the frequency of words in a post, and are primarily responsible for propagating features across the topic/propagation tree. Similarly, in order to avoid the excessive propagation of features caused by the number of *word-post* edges in the post-level heterogeneous graph, we also calculate and filter the weights of such edges. Specifically, the relationship between a word node w_k and a post node $p_{i,j}$ is jointly determined by the number of times that w_k appears in $p_{i,j}$ and in the whole corpus (i.e., Term Frequency−Inverse Document Frequency (TF-IDF)). In order to avoid extra statistical operations, we improve the calculation method of TF-IDF with PMI statistics:
  $$\mathrm{TF-IDF}(w_k, p_{i,j}) = \frac{N_{w_k}}{N_{w_k,p_{i,j}}} \times \log \frac{N_{window}}{\{n : w_k \in window\} + 1}, \tag{3}$$
  where $N_{w_k,p_{i,j}}$ is the number of windows that w_k appears in $p_{i,j}$, and $\{n : w_k \in window\}$ is the number of windows in the corpus that w_k appears.
- *post-post*
  The relationship between the post nodes p_i and p_j is determined by the given propagation tree structure, and its strength is calculated as:
  $$\begin{aligned}\mathrm{Strength}(p_{i,m}, p_{i,n}) &= -\log Similarity_{(p_{i,m},p_{i,n})}, \\ &= -\log(h_{p_{i,m}} \cdot h_{p_{i,n}}^{\top}), \end{aligned} \tag{4}$$
  where $h_{p_{i,m}}$ is the vector representation of $p_{i,m}$. In particular, only such edges are asymmetric in the adjacency matrix.

As illustrated in Figure 4, all the posts are included in the post-level heterogeneous graph, and each node could acquire interaction information from its neighbors. Due to adding word nodes as a bridge, the post nodes can obtain semantic information across rumors. Since the structure conforms to the standard hypothesis and conditioned adjacency matrix of the graph allows the model to distribute gradient information from the supervised loss for learning representations of nodes both with and without labels [27], [28], the stance detection task can be framed as a node classification task. Similarly, in the post-level heterogeneous graph, the rumor detection task can be defined as a sub-graph classification task. On a higher-order graph, rumors and words are treated as nodes, and the basic assumption of label propagation is still met, so we define it as a classification task on the rumor-level graph.

Different from the post-level heterogeneous graph, the rumor-level heterogeneous graph contains only two kinds of nodes and three kinds of edges. Among them, the nodes are *word* nodes (directly inherited from the post-level heterogeneous graph), and *rumor* nodes (generated by graph pooling operations). Both *word-rumor* edges, *rumor-rumor* and *word-word* edges are partly inherited from the post-level heterogeneous graph, and the weights are also retained.

C. Multilayer Graph Neural Networks (Multi-GNNs)

After the construction of hierarchical heterogeneous graph, we feed it into multilayer GNNs and complete stance and rumor detection tasks in different stages. As illustrated in Figure 4, the multilayer neural networks introduced can be divided into two stages:

1) Post-Level Heterogeneous Graph: As the interaction between posts in rumors can guide the judgment of posts and rumors, previous work tries to realize the representation learning of posts by adjusting the weights between posts (e.g., graph convolution or graph attention), and extracting structure and attribute characteristics [21], [23]. In contrast, we set up new connections between posts in different rumor/topic in the post-level heterogeneous graph. However, existing methods use fixed order or specific local post pair to build the neighborhood, which cannot capture rich, high-order structural details. To solve this problem, we propose Random Walk with Restart (RWR) [53] commonly used in information retrieval to explore global topology and obtain adaptive neighbors by

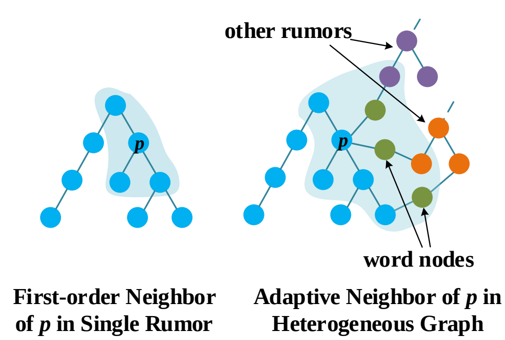


Fig. 5. Adaptive neighborhood partitioning.

iteratively moving, as shown in Figure 5. Specifically, we propose an Adaptive Graph ATtention (AGAT) layer to learn the representation of each post.

Consider a RWR on a post i, with p_i and adjacency matrix A_i. The particle starts from the center p_i and randomly walks to its neighbors in G with a probability proportional to edge weights. In each step, it also has a certain probability to return to the center node. The iteration can be written as

$$\mathrm{w}_i^{(t+1)} = c \cdot \widetilde{A}_i \mathrm{w}_i^{(t)} + (1-c) \cdot \mathrm{e}_i, \tag{5}$$

where $\widetilde{A}_i$ is the transition probability matrix by normalizing columns of A_i, $c \in [0,1]$ is a trade-off parameter between RWR, e_i is a vector of all zeros except the entry corresponding to the center post p_i, and w_i quantifies the proximity between the center post p_i and all other nodes.

$h_i = [h_{i,sem}; h_{i,att}] \in H$ is the feature of node i in G, where $h_{i,sem}$ and $h_{i,att}$ are the corresponding semantic and post attribute information. For capturing the information of the node's adaptive neighbors while preserving its properties, we introduce original Graph ATtention (GAT) to update p_i [28]. For one layer GAT, we perform self-attention by computing attention coefficients

$$\alpha_{ij} = \frac{\exp(\mathbf{LeakyReLU}(a[\mathbf{W}h_i; \mathbf{W}h_j]))}{\sum_{k\in|H_i|}\exp(\mathbf{LeakyReLU}(a[\mathbf{W}h_i; \mathbf{W}h_k]))}, \tag{6}$$

where $\mathbf{W}$ is weight matrix, a is random vector, H_i is the set of adaptive neighbors, and $\mathbf{LeakyReLU}(\cdot)$ is a nonlinear activation function. To stabilize the learning process of self-attention, we employ multi-head attention to sum multiple features as $k+1$th layer outputs

$$h_i^{(k+1)} = \sigma(\sum_{j\in|H_i|} \alpha_{ij} \mathbf{W} h_j^{(k)}). \tag{7}$$

To obtain the node embedding for learning rumor representation, we retain the dimension of $H^{(k)}$, and use a dense layer and softmax function for stance classification:

$$Y = \mathrm{softmax}(\mathrm{FC}(H^{(k)})), \tag{8}$$

where $\mathrm{softmax}(x) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$, $\mathrm{FC}(\cdot)$ is the fully connected layer. Then, we can minimize the cross-entropy loss for labeled data distribution for model training:

$$\mathcal{L} = -\sum_{p_{i,j}\in P_{labeled}} \sum_{l\in L} \widehat{Y}_{p_{i,j},l} \ln Y_{p_{i,j},l}, \tag{9}$$

where $P_{labeled}$ is the set of node indices that have labels, L is the label set, and $\widehat{Y}$ is the corresponding label indicator matrix.

2) Rumor-Level Heterogeneous Graph: To further learn rumor representation, we propose a structure-sensitive Graph Pooling Layer (GPL) to capture both semantic and structural (i.e., interaction) features. As shown in Figure 4, the GPL coarsens the post-level heterogeneous graph to a rumor-level heterogeneous graph, and we obtain an initial representation of each rumor node from the representation of the corresponding post subgraph, the pseudocode is shown in Algorithm 1.

Specifically, we use a trainable assignment matrix S to implement the graph pooling. During the training process, the assignment matrix S is driven by two downstream classification tasks to automatically combine the semantic features of the nodes and capture structure information of each subgraph (i.e., rumor). To be more sensitive to user interactions, we update $\widetilde{A}_s$ by combining weights from interaction types, i.e., introducing the ratio matrix M of the interaction type of corresponding post pair to all interaction types. As shown in Figure 3, posts with clear attitudes are more valuable to distinguish the veracity of rumors, thus we refer to the calculation method of TF-IDF to calculate the weight of post pairs in different rumors. In other words, we regard a rumor as a document, so 'support-deny', 'query-deny', and all rest of the interactions are equal to different words. Then, each element of ratio matrix $M_{i,j}$ can be calculated as follow

$$\begin{aligned} M_{i,j} &= TF_{i,j} \times IDF_{type_{i,j}}, \\ &= \frac{n_{type_{i,j}}}{n_{r_m}} \times \log(\frac{n_{rumor}}{\{n_{type} : (type_{i,j}) \in r_{type}\}}), \end{aligned} \tag{10}$$

where $type_{i,j}$ is the interaction type of post pair (p_i, p_j), $n_{type_{i,j}}$ denotes the number of interaction type $type_{i,j}$ in rumor r_m, n_{r_m} is the number of post pairs in rumor r_m, n_{rumor} is the number of all rumors in the dataset, and $\{n_{type} : (type_{i,j}) \in r_{type}\}$ denotes the number of rumors that contain interaction type $type_{i,j}$ in the dataset.

As illustrated in Figure 4, in the rumor-level heterogeneous graph, the coarsened rumor node inherits the relationship between the original post node and word node. To capture the features of the rumor node neighbors as well, we stack 2-layer AGAT behind the GPL. Since the loss and model parameter updating is similar to AGAT, we omit it here.

D. Joint Training

Since the existing multi-task learning methods have indicated that the joint post stance and rumor veracity detection can effectively improve the performance of both two tasks, we also combined the two tasks in this work. However, different from the existing method of sharing components, we construct a serial graph neural network model. Among them, each epoch contains all training samples in both downstream classification tasks, and the loss of the two classifiers is shared by the complete model. In each complete training batch, we adjust the unbalanced label distribution on the one hand to ensure the training sample balance (i.e., making sure each label has the same number of samples by repeating the fewer label samples); On the other hand, on the basis of ensuring the same number of classified samples, the losses provided by the two classifiers shown in Figure 4 are accumulated to realize joint training.

IV. EXPERIMENTS

In this section, we first discuss the model performance of the proposed joint detection model in a dataset for both tasks. Furthermore, we analyze the performance gap between the two independent models and the joint model, and then discuss the changes brought by the joint detection. Finally, we conduct some experiments of the proposed Multi-GNNs from several independent perspectives, and analyze the advantages and interpretation of the model.

Algorithm 1: Structure Sensitive Graph Pooling

Data: Node embedding and adjacency matrix of post-level heterogeneous graph: H' and $\widetilde{A}$, weight proportion: α, ratio matrix of the interaction type: $M \in \mathbb{R}^{n \times n}$, assignment matrix: $S \in \mathbb{R}^{n' \times n}$.
Result: Initial node embedding and adjacency matrix of rumor-level heterogeneous graph: H'_{pool} and $\widetilde{A}_{pool}$.

```
Initialize S and M;
for i ← 1 to n do
    for j ← 1 to n do
        if Ã_ij ≠ 0 then
            M_ij ← the ratio of the number of interaction types corresponding to p_i, p_j.
        end
    end
end
Ã_s ← αÃ + (1 − α)M // Updating the original adjacency matrix. α is a parameter that adjusts the ratio of M to A.
    When α = 0, the input matrix is the original adjacency matrix A. ;
H'_pool ← SH' // Generating the rumor-level feature matrix from the post-level feature matrix. ;
Ã_pool ← SÃ_sS^⊤ // Generating the rumor-level adjacency matrix from the post-level adjacency matrix.
```

A. Datasets and Evaluation Metrics

Since our method conducts both the stance detection and rumor detection tasks, we run our experiments on task7 of SemEval2019, which is a widely used benchmark from PHEME[1]. The rumors come from real social networks: Twitter and Reddit, where each rumor consists of a propagation tree in which each post has several attribute tags. The dataset of RumourEval 2019 is summarized in Table I and Table II. Due to the extremely unbalanced distribution of samples in the dataset, the evaluation metric accuracy cannot reflect the feature learning ability of the model[2]. Hence, we use Macro-averaged F1 scores as evaluation metrics for both tasks.

First, we conduct a joint detection of post stance and rumor veracity in the dataset, and we will discuss the experiments of the two tasks respectively in the following.

B. Stance Classification

Stance detection can be regarded as a classic text classification task. We will compare the joint detection methods and some common sequence models.

1) Baseline Methods: We compare our method with multiple state-of-the-art stance detection models as follows:

- **Major Vote**: This method intuitively takes the category of majority in the training set to predict the data in the test set.

[1]https://www.pheme.eu/

[2]Our method requires the dataset to contain both rumor and stance detection in the same rumor.

- **LR**: A machine learning method (i.e., Logistics Regression [6]) utilize various manual features and carefully adjust category weights.
- **MT-ES, MTL2, VRoC**: They are multi-task learning methods with enhanced shared layer [9], shared LSTM layer [13], and shared encoder [30]. The shared components in those models are usually trained together by the losses from two or more tasks. The shared component has been proven to extract more stable and efficient features to improve the performance and stability of the model for detecting the rumors, the stance of posts and etc.
- **Branch-LSTM**: Branch-LSTM is a LSTM based sequence labeling model [17] for stance classification. Branch-LSTM can encode the given sequence information to obtain high-quality under the premise of considering global information, then achieve stance classification.
- **TreeLSTM**: It is a Tree LSTM Model equipped with child convolve and max-pooling cell [11]. Specifically, this method realizes local information extraction by introducing a convolutional encoder adapted to the tree structure, and finally realizes stance classification by pooling operation.
- **CLEARumor**: CLEARumor is a CNN based neural network equipped with hand-crafted features [18]. This method defines a large number of interpretable features to transform input information into high-quality vectorization features, and then utilizes a CNN component to

TABLE I
STATISTICS OF STANCE DETECTION.

		Support	Deny	Query	Comment	Total
Train	Twitter	910	344	358	2907	5217
	Reddit	15	34	37	612	
Dev	Twitter	94	71	106	778	1485
	Reddit	8	11	14	403	
Test	Twitter	141	92	62	771	1827
	Reddit	16	9	31	705	

TABLE II
STATISTICS OF RUMOR DETECTION.

		True	False	Unverified	Total
Train	Twitter	137	62	98	327
	Reddit	7	17	6	
Dev	Twitter	8	12	8	38
	Reddit	2	7	1	
Test	Twitter	22	30	4	81
	Reddit	9	10	6	

TABLE III
THE RESULTS OF STANCE DETECTION.

Methods	MacF1	S F1	D F1	Q F1	C F1
Major Vote	21.0	0	0	0	84.1
LR	52.4	43.5	21.5	56.1	88.5
MT-ES	48.9±0.4	35.3±0.5	18.8±0.5	51.6±0.3	86.2±0.3
Branch-LSTM	50.1±0.3	39.5±0.3	20.8±0.6	54.8±0.3	85.0±0.2
CLEARumour	50.8±0.5	41.7±0.5	21.0±0.7	54.1±0.3	86.4±0.3
MTL2	52.3±0.5	42.9±0.5	21.7±0.5	57.5±0.4	87.0±0.3
VRoC	54.8±0.3	46.8±0.4	24.5±0.5	60.0±0.3	**88.7±0.2**
TreeLSTM	53.0±0.5	44.7±0.5	22.1±0.4	57.4±0.4	87.8±0.3
Conversational-GCN	54.1±0.5	45.9±0.4	23.6±0.7	59.1±0.4	87.7±0.3
Multi-GNNs(GCN)	54.5±0.4	45.6±0.3	23.7±0.5	60.6±0.4	87.9±0.3
- without rumor loss	53.2±0.5	45.2±0.3	22.9±0.6	60.0±0.4	86.8±0.3
Multi-GNNs(GAT)	55.1±0.4	46.6±0.3	24.2±0.4	61.7±0.3	87.9±0.2
- without rumor loss	54.4±0.4	45.9±0.5	23.5±0.6	61.2±0.3	87.2±0.2
Multi-GNNs(AGAT)	**55.7±0.3**	**47.2±0.3**	**24.8±0.4**	**62.3±0.3**	88.5±0.3
- without rumor loss	55.1±0.4	46.4±0.4	24.4±0.5	61.8±0.3	88.1±0.2

realize the stance detection task.

- **Conversational-GCN**: It is a method modeling structure using GCN [21]. This method dynamically models the sequential propagation of posts and utilizes GCN to extract local features.
- **Multi-GNNs(GCN), Multi-GNNs(GAT)**: They are the GCN and GAT version of this work (removing RWR). In other words, these methods do not have the ability to adjust its neighbor partition adaptively.
- **Multi-GNNs(AGAT)**: Multi-GNN is the proposed method in this paper. Multi-GNNs(only BOW) utilizes word bag features of high-frequency words as post initialization.

2) Experiment Setup and Results Analysis: To compare the effects of the model fairly, we equipped the deep learning methods with the same initialization features, i.e., the 512 dimension word embedding and the 246 dimension attribute features. In Multi-GNNs, we stack 2-layer AGATs and one fully connected layer, and the input and output dimensions of the middle AGAT layer are consistent. We set $c = 0.5$ in RWR, set the number of attention heads as 8, and the rest remain at the default settings. We trained our model for a maximum of 300 epochs using Adam optimizer [54].

The experimental results are shown in Table III, which indicates that the methods based on deep neural networks and multi-task training have high performance. Furthermore, our model significantly outperforms baseline methods 1-2% in the stance detection task. From a more detailed perspective, our model is robust in extremely unbalanced datasets without weight settings (e.g., LR) and shared components (e.g., MT-ES, MTL2, and VRoC). It is because our method fits the pattern of stance interaction by capturing the distribution features of the target post neighbors, and we introduce the commonality extraction between topics through the high-frequency words as the bridge. Particularly, the information provided by the adaptive neighbor is richer and more efficient through different methods of neighborhood division (i.e., Conversational-GCN and Multi-GNNs(GCN/GAT/AGAT)). In addition, compared with the performance of the model with only reserved stance detection loss (i.e., - without rumor loss), we find that the combined rumor detection loss can improve the stance classification accuracy by almost 1%.

C. Rumor Detection

Rumor detection can be regarded as a propagation tree classification task. We will compare the joint detection methods and some common multi-task learning models.

1) Baseline Methods: We select the following rumor detection models as baselines:

- **MT-ES, VRoC**: They are both the multi-task learning methods [9], [30] with enhanced shared layer and shared encoder. In this paper, the shared components and according specific task layers in both models are trained by rumor detection and stance detection respectively. In particular, unlike the models that have only individually shared components, this kind of model needs to be equipped with a specific task layer.
- **CLEARumor**: A MLP based classifier equipped with hand-crafted features [18]. Similar to the stance classification, CLEARumor defines more complex features for the veracity classification of rumors and uses multi-layer neural networks to achieve final label.
- **MTL2, TreeLSTM, Hierarchical-PSV**: The same components are shared with stance detection, and a specific classifier based on multi-task learning is provided for rumor detection [11], [13], [21]. In this work, multi-task learning is defined only by stance detection and rumor detection. In practice, several other tasks have also been shown to improve performance by incorporating them into multi-task learning, e.g., rumor tracing.
- **Bi-subgraphGAT**: A subgraph attention method using user and word heterogeneous graphs [55]. Bi-

TABLE IV
THE RESULTS OF RUMOR DETECTION.

Method	MacF1	RSME
MT-ES	27.8±0.6	0.8771
CLEARumor	29.3±0.6	0.8463
TreeLSTM	33.5±0.4	0.8301
Bi-subgraphGAT	35.7±0.5	0.8144
TD-RvNN-GA	36.1±0.5	0.8104
MTL2	30.7±0.6	0.8411
Hierarchical-PSV	36.2±0.4	0.8093
VRoC	35.1±0.4	0.8182
Multi-GNNs(GCN)	35.3±0.4	0.8167
- without stance loss	33.4±0.5	0.8355
Multi-GNNs(GAT)	35.9±0.5	0.8128
- without stance loss	33.6±0.5	0.8319
Multi-GNNs(AGAT)	**36.7±0.3**	**0.8012**
- without stance loss	34.0±0.4	0.8287

subgraphGAT introduces users to construct heterogeneous graphs, however, this change is on the one hand limited to whether the user data is provided or not, on the other hand, the ability to promote label propagation is weaker than that of words.

- **TD-RvNN-GA**: An extension model of TD-RvNN with global attention [23].
- **Multi-GNNs(GCN), Multi-GNNs(GAT)**: They are the GCN and GAT version of our work.
- **Multi-GNNs(AGAT)**: Multi-GNNs is the proposed method inherited node embedding from stance detection while stacking GPL and AGAT.

2) Experiment Setup and Results Analysis: In this task, we follow the evaluation metrics in the previous section, and set the weight proportion $\alpha = 0.5$ in GPL. Because the rumor veracity is difficult to determine, evaluating the gap between the prediction possibility and the real veracity can more accurately assess the models. Hence, we utilized the RSME metric to count the difference between them. The experimental results are shown in Table IV, we can find that our method is better than the other multi-task methods or attention based methods. Especially, our veracity estimation variance (i.e., RSME) is much smaller than other methods. Besides, the performance difference between Multi-GNNs(GCN/GAT) and Multi-GNNs(AGAT) also indicates that the adaptive neighbor through RWR can remarkably improve the effect of rumor detection. It is mainly because the GPL and the stacked AGAT layers can capture more valuable structure, semantics, and target neighbor distribution features driven by classification loss. In addition, we also tested the performance of the model without the stance classification loss and found that it was significantly lower than the results after the joint training. It indicates that the stance classification loss introduced by the joint training is more obvious in improving the performance of rumor detection.

D. Independent Analysis

In this section, we further explore the performance of joint detection on the graph model. Specifically, we will conduct independent experiments for two tasks in a larger dataset and discuss the performance of two independent tasks respectively.

a) Independent Stance Detection: We conduct experiments only on the post-level heterogeneous graph (i.e., removing the loss from rumor detection). The dataset utilized in this independent experiment is PHEME, and SemEval2019 is a subset of PHEME. We remain the experimental settings as Section IV, and the experimental results are shown in Table. We can find that our model based on heterogeneous graphs can still obtain comparable results when only uses stance classification loss. At the same time, the variances obtained by our independent model over 10 trials were also slightly lower than those obtained by other baseline methods, indicating that it was able to capture more robust features.

b) Independent Stance Detection: Similarly, we also conduct an extra experiment in rumor detection, i.e., removing the loss from stance detection but retaining the hierarchical heterogeneous structure and embedding process. Meanwhile, to test the performance of our method in large-scale datasets, we chose two larger datasets and refer to their setting, as shown in Table VI. The experimental results are shown in Table VII and Table VIII, some of the baseline methods used for comparison are directly copied from the recent work (marked with), and the remaining baseline methods come from running code (based on adjusted parameters). We can observe that the methods based on deep learning (e.g., Bi-GCN, RvNN, and TD-RvNN) are better than those based on hand-crafted features (e.g., DTC and SVM-TK). Meanwhile, the results indicate that the method based on multi-task learning achieves high performance, but our model can still achieve the state-of-the-art result when missing the loss of stance detection.

E. Further Discussion

In order to analyze the performance of Multi-GNNs from more perspective, we will conduct more experiments to analyze its performance in a few size samples, early detection and other situations.

1) Size of Labeled Data: In the real environment, rumors spread quickly and widely, easily causing a negative impact on social media. At the same time, it costs a lot to obtain the labeled data of rumor veracity, and its timeliness and novelty make it unable to use the existing labeling data. Therefore, the requirement of labeled data is an extremely important evaluation metric for the models. To evaluate the effect of the size of the labeled data, we compare our method with baselines with different proportions of the training data in both stance and rumor detection tasks. The experimental results are shown in Figure 6, which shows that our method still maintains high performance in different proportions of training data. Especially when the proportion of labeled data is less than 0.4, our method is significantly better than other baseline methods. It indicates that our architecture can effectively capture the semantics and the local structural features, and transform them

TABLE V
THE RESULTS OF INDEPENDENT STANCE DETECTION.

Methods	MacF1	S F1	D F1	Q F1	C F1
MT-ES	43.6±0.5	31.5±0.3	15.3±0.6	**53.4±0.4**	**74.1±0.5**
Branch-LSTM	49.5±0.4	42.8±0.4	38.5±0.5	48.1±0.3	68.3±0.4
CLEARumour	49.9±0.5	42.9±0.5	38.9±0.5	48.7±0.4	68.8±0.4
MTL2	50.9±0.4	43.7±0.5	40.1±0.45	49.8±0.6	70.2±0.3
VRoC	**52.5±0.4**	**45.6±0.5**	41.4±0.4	51.5±0.3	71.5±0.3
TreeLSTM	50.6±0.4	43.4±0.5	40.4±0.5	49.0±0.4	69.7±0.4
Conversational-GCN	52.0±0.4	44.8±0.3	41.1±0.6	51.3±0.4	70.7±0.3
Multi-GNNs(GCN)	51.0±0.4	41.2±0.4	40.7±0.7	51.0±0.3	67.9±0.3
Multi-GNNs(GAT)	51.3±0.4	44.9±0.5	41.1±0.6	51.2±0.3	68.1±0.2
Multi-GNNs(AGAT)	**52.5±0.3**	45.4±0.4	**42.4±0.5**	52.4±0.3	70.4±0.2

TABLE VI
STATISTICS OF THE TWITTER DATASETS.

	Twitter15	Twitter16
# of posts	331,612	204,820
# of users	276,663	173,487
# of events	1,490	818
# of true rumors	374	205
# of false rumors	370	205
# of unverified rumors	374	203
# of non-rumors	372	205
# of avg. # of posts/events	223	251
# of max. # of posts/events	1,768	2,765
# of min. # of posts/events	55	81

TABLE VII
THE RESULTS OF INDEPENDENT RUMOR DETECTION ON TWITTER 15.

Methods	Acc.	N F1	F F1	T F1	U F1
DTC	0.454	0.415	0.355	0.733	0.317
SVM-TK	0.750	0.804	0.698	0.765	0.733
RvNN	0.723	0.682	0.758	0.821	0.654
Bi-GCN	0.886	0.891	0.860	0.930	0.864
TD-RvNN	0.887	0.893	0.861	0.929	0.862
VRoC	0.899	0.903	0.884	0.934	0.873
Ours(GCN)	0.891	0.894	0.880	0.928	0.867
Ours(GAT)	0.895	0.901	0.877	0.930	0.881
Ours(AGAT)	**0.901**	**0.905**	**0.885**	**0.942**	**0.884**

TABLE VIII
THE RESULTS OF INDEPENDENT RUMOR DETECTION ON TWITTER 16.

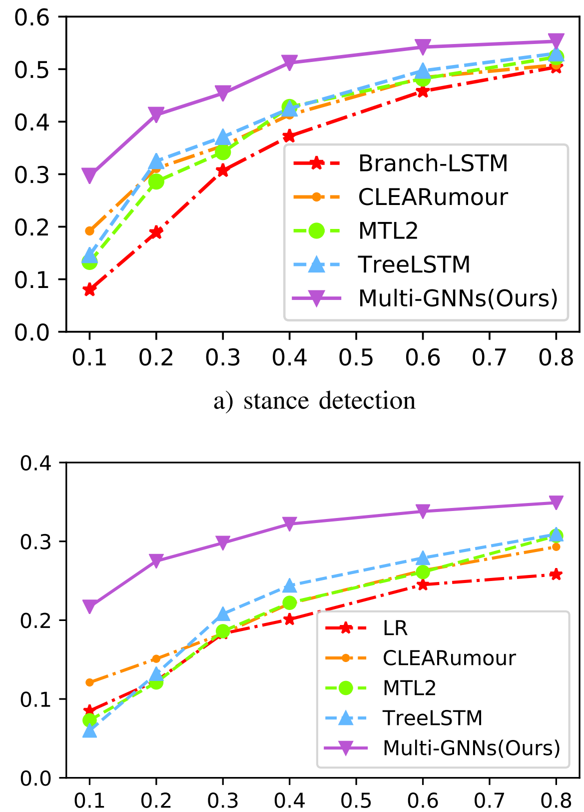
Methods	Acc.	N F1	F F1	T F1	U F1
DTC	0.477	0.254	0.080	0.190	0.482
SVM-TK	0.735	0.740	0.709	0.836	0.686
RvNN	0.739	0.662	0.743	0.835	0.708
Bi-GCN	0.883	0.847	0.869	0.937	0.865
TD-RvNN	0.889	0.889	0.857	0.928	0.864
VRoC	0.896	0.892	0.879	0.927	0.864
Ours(GCN)	0.892	0.894	0.884	0.930	0.866
Ours(GAT)	0.895	0.900	0.891	0.935	0.869
Ours(AGAT)	**0.900**	**0.904**	**0.896**	**0.940**	**0.874**

to other nodes on the same graph. This characteristic can effectively alleviate the dependence of stance and rumor detection on the size and topic of the labeled data, and facilitate its wide application in the complex and varied Internet environment.

2) Early Rumor Detection: Due to the terrible consequences of the diffusion of rumors, much research has focused on early rumor detection. In this work, we define the different stages of rumor detection according to the different proportions of the depth of the rumor propagation tree or time series. The results of the experiment are shown in Figure 7. Since some methods do not use propagation structure, we choose four methods for comparison. We can observe that in the early stage of rumor diffusion, all methods can only rely on the rumor's semantics or a small amount of interaction to judge its veracity, so the performance is significantly reduced except for the proposed methods. However, the method based on hierarchical heterogeneous graphs can better adapt to the requirements of early detection than other methods, and quickly restore the detection performance in the middle of rumor propagation. The reason is that our model connects rumors of different topics, and can still be judged by the topic and neighbor distribution characteristics when the propagation structure is insufficient.

3) Visualization: In order to further observe the changes of initial post features and post features of AGAT output, we plan to realize the visualization of post features through dimensionality reduction to observe the changes of post-distribution. By using the original dimension reduction method, we respectively visualize the post representation obtained by AGAT to observe the effect of GNNs intuitively. As shown in Figure, we find that the initial node features are not well distributed according to the categories, and the post features

4) Interpretation: As a bridge to link posts and realize feature propagation, high-frequency words are very important components in constructing hierarchical heterogeneous graphs. Therefore, we want to further explore its role in detecting the stance of posts and the veracity of rumors. Specifically, because in post-level and rumor-level heterogeneous graphs,



a) stance detection

b) rumor detection

Fig. 6. The results of the models are trained using different proportions of data in training set, where the x-axis represents the proportion of training data and the y-axis represents the MacF1.

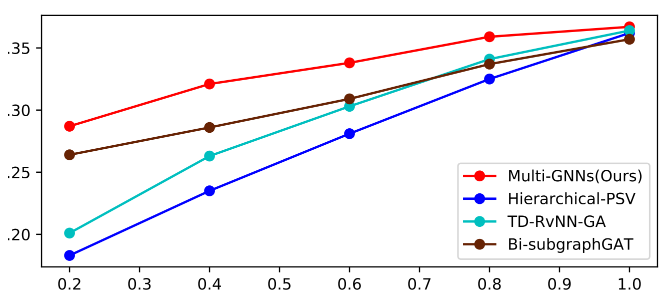
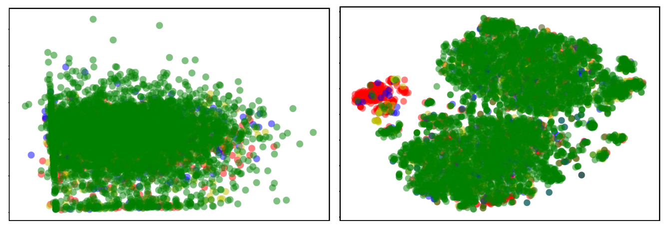


Fig. 7. The detection effect of the models in different stages of rumor propagation, where the x-axis represents the residual proportion of the propagation tree and the y-axis represents the MacF1.

the high-frequency word node can be treated as one-word post and one-post propagation trees, respectively, we regard the word nodes in different heterogeneous graphs as posts and rumors respectively and use their representations to carry out stance and rumor classification. We select the top-5 of the score of each category after classification, as shown in Table IX. We find that the words in ‘Support’ are related to news or report, the words in ‘Deny’ and ‘Query’ have obvious corresponding emotional colors, and only the ‘Comment’ is difficult to distinguish. In the rumor-level heterogeneous graph,



a) input node features b) output node features

Fig. 8. Visualization results of model input features and output embeddings. The correspondence between color and label is Comment, Support, Deny, and Query.

TABLE IX
TOP-5 WORDS IN DIFFERENT STANCE AND DIFFERENT VERACITY OF RUMOR.

Support	latest, released, news, confirmed, reports
Deny	believe, wtf, really, stop, stupid
Query	if, how, facts, actually, lies
Comment	know, police, killed, people, nothing
True	reports, latest, hostages, confirmed, release
False	shooting, robbery, armed, gunman, crashed
Unverified	attack, soldier, source, fired, found

we observe that some words with positive and expressive tone were marked as true, while some words about attack were marked as false, which points out the difference between the descriptions of different veracity rumors. These phenomena, on the one hand, illustrate the semantic characteristics of different rumors and posts in a specific category, and on the other hand, show that our method can learn interpretable node embedding.

V. CONCLUSION

We jointly define stance detection and rumor detection on a hierarchical heterogeneous graph, and propose a set of GNNs to accomplish two tasks simultaneously. Our method is a novel model, which can realize the sharing of annotation information between different topics by constructing hierarchical heterogeneous graphs. On the one hand, it can improve the detection performance, and on the other hand, it can realize the goal of early detection and reduce the consumption of annotation. Experimental results on real datasets show that our method can effectively improve the detection effect. Meanwhile, some additional experiments indicate that our method is more efficient than other methods and does not rely on the quantity of labeled data.

ACKNOWLEDGEMENT

This work was supported by the NSFC program (No. U20B2053, 62002007, and 61772151), S&T Program of Hebei through grant 20310101D, and SKLSDE-2020ZX-12. Lifang He and Lichao Sun were supported by NSF ONR N00014-18-1-2009 and Lehigh’s accelerator grant S00010293. Philip S. Yu was supported by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

REFERENCES

[1] H. Peng, J. Li, Y. Song, R. Yang, R. Ranjan, P. S. Yu, and L. He, "Streaming social event detection and evolution discovery in heterogeneous information networks," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 5, pp. 89:1–89:33, 2021.

[2] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," *CoRR*, vol. abs/2007.02500, 2020.

[3] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 2016, pp. 3818–3824.

[4] P. Donvan, "How idle is idle talk? one hundred years of rumor research," *Diogenes*, vol. 54(1), pp. 59–82, 2007.

[5] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 598–608, 2017.

[6] B. Ghanem, A. T. Cignarella, C. Bosco, P. Rosso, and F. M. R. Pardo, "UPV-28-UNITO at semeval-2019 task 7: Exploiting post's nesting and syntax information for rumor stance classification," in *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, 2019, pp. 1125–1131.

[7] 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, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 708–717.

[8] ——, "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, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 1980–1989.

[9] ——, "Detect rumor and stance jointly by neural multi-task learning," in *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, 2018, pp. 585–593.

[10] H. Guo, J. Cao, Y. Zhang, J. Guo, and J. Li, "Rumor detection with hierarchical social attention network," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 2018, pp. 943–951.

[11] Q. Li, Q. Zhang, and L. Si, "Rumor detection by exploiting user credibility information, attention and multi-task learning," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2019, pp. 1173–1179.

[12] S. Kumar and K. M. Carley, "Tree lstms with convolution units to predict stance and rumor veracity in social media conversations," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2019, pp. 5047–5058.

[13] E. Kochkina, M. Liakata, and A. Zubiaga, "All-in-one: Multi-task learning for rumour verification," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, 2018, pp. 3402–3413.

[14] S. Qu, Z. Zhao, L. Fu, X. Wang, and J. Xu, "Joint inference on truth/rumor and their sources in social networks," in *39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*, 2020, pp. 924–933.

[15] Z. Lingyu, W. Bin, and W. Bai, "Detecting rumor on microblogging platforms via a hybrid stance attention mechanism," in *Web Engineering - 20th International Conference, ICWE 2020, Helsinki, Finland, June 9-12, 2020, Proceedings*, 2020, pp. 347–364.

[16] A. Janchevski and S. Gievska, "Andrejjan at semeval-2019 task 7: A fusion approach for exploring the key factors pertaining to rumour analysis," in *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, 2019, pp. 1083–1089.

[17] E. Kochkina, M. Liakata, and I. Augenstein, "Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm," in *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, 2017, pp. 475–480.

[18] I. Baris, L. Schmelzeisen, and S. Staab, "Clearumor at semeval-2019 task 7: Convolving elmo against rumors," in *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, 2019, pp. 1105–1109.

[19] Z. Lingyu, S. Chenguang, W. Bin, and W. Bai, "SMAM: detecting rumors from microblogs with stance mining assisting task," in *Fourth IEEE International Conference on Data Science in Cyberspace, DSC 2019, Hangzhou, China, June 23-25, 2019*, 2019, pp. 242–249.

[20] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, 2016, pp. 31–41.

[21] P. Wei, N. Xu, and W. Mao, "Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2019, pp. 4786–4797.

[22] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020, pp. 549–556.

[23] J. Ma, W. Gao, S. R. Joty, and K. Wong, "An attention-based rumor detection model with tree-structured recursive neural networks," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 4, pp. 42:1–42:28, 2020.

[24] L. M. S. Khoo, H. L. Chieu, Z. Qian, and J. Jiang, "Interpretable rumor detection in microblogs by attending to user interactions," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020, pp. 8783–8790.

[25] B. Rath, W. Gao, J. Ma, and J. Srivastava, "From retweet to believability: Utilizing trust to identify rumor spreaders on twitter," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, Sydney, Australia, July 31 - August 03, 2017*, 2017, pp. 179–186.

[26] H. Chen, B. Chen, C. Ai, M. Zhu, K. Lai, and L. He, "Research on rumor propagation simulation based on behavior-attribute," in *5th IEEE International Conference on Data Science in Cyberspace, DSC 2020, Hong Kong, July 27-30, 2020*, 2020, pp. 225–231.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[28] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *CoRR*, vol. abs/1710.10903, 2017.

[29] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," *CoRR*, vol. abs/1809.05679, 2018.

[30] M. Cheng, S. Nazarian, and P. Bogdan, "Vroc: Variational autoencoder-aided multi-task rumor classifier based on text," in *WWW*, 2020, pp. 2892–2898.

[31] H. Peng, R. Yang, Z. Wang, J. Li, and R. Ranjan, "Lime: Low-cost and incremental learning for dynamic heterogeneous information networks," *IEEE Transactions on Computers*, vol. PP, no. 99, 2021.

[32] Y. Hei, R. Yang, H. Peng, L. Wang, X. Xu, J. Liu, H. Liu, J. Xu, and L. Sun, "HAWK: rapid android malware detection through heterogeneous graph attention networks," *CoRR*, vol. abs/2108.07548, 2021.

[33] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, Eds. ACM, 2011, pp. 675–684.

[34] F. Yang, X. Yu, Y. Liu, and M. Yang, "Automatic detection of rumor on sina weibo," in *KDD Workshop*, 2012.

[35] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *IEEE Trans. Multim.*, vol. 19, no. 3, pp. 598–608, 2017.

[36] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, A. Gangemi, S. Leonardi, and A. Panconesi, Eds. ACM, 2015, pp. 1395–1405.

[37] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*,

J. Gehrke, W. Lehner, K. Shim, S. K. Cha, and G. M. Lohman, Eds. IEEE Computer Society, 2015, pp. 651–662.

[38] N. Rosenfeld, A. Szanto, and D. C. Parkes, "A kernel of truth: Determining rumor veracity on twitter by diffusion pattern alone," in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2020, pp. 1018–1028.

[39] J. Ma, W. Gao, and K. Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning," in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2019, pp. 3049–3055.

[40] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang, "Rumor detection on social media with graph structured adversarial learning," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2020, pp. 1417–1423.

[41] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, "Jointly embedding the local and global relations of heterogeneous graph for rumor detection," in *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*. IEEE, 2019, pp. 796–805.

[42] M. Lukasik, K. Bontcheva, T. Cohn, A. Zubiaga, M. Liakata, and R. Procter, "Gaussian processes for rumour stance classification in social media," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 20:1–20:24, 2019.

[43] M. Lukasik, P. K. Srijith, D. Vu, K. Bontcheva, A. Zubiaga, and T. Cohn, "Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016.

[44] Y. Chen, Z. Liu, and H. Kao, "IKM at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor verification," in *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, 2017, pp. 465–469.

[45] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, 2018, pp. 1063–1072.

[46] Y. Zhang, Q. Liu, and L. Song, "Sentence-state LSTM for text representation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 317–327.

[47] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2018, pp. 3546–3553.

[48] Y. Huang, P. Peng, Y. Jin, Y. Li, and J. Xing, "Domain adaptive attention learning for unsupervised person re-identification," in *AAAI*, 2020, pp. 11 069–11 076.

[49] K. Zhang, Y. Zhu, J. Wang, and J. Zhang, "Adaptive structural fingerprints for graph attention networks," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[50] Z. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, 2018, pp. 4805–4815.

[51] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 3734–3743.

[52] G. Gorrell, A. Aker, K. Bontcheva, L. Derczynski, E. Kochkina, M. Liakata, and A. Zubiaga, "Semeval-2019 task 7: Rumoureval, determining rumour veracity and support for rumours," in *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, 2019, pp. 845–854.

[53] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *ICDM*. IEEE, 2006, pp. 613–622.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[55] Q. Huang, J. Yu, J. Wu, and B. Wang, "Heterogeneous graph attention networks for early detection of rumors on twitter," *CoRR*, vol. abs/2006.05866, 2020.

Chen Li Chen Li is a Ph.D. candidate in the Department of Computer Science and Engineering, Beihang University, China. He received the MS degree in Communication University of China in Beijing, China and received the BS degree in Nankai University in Tianjin, China. His current research interests include graph neural network, machine learning, and data mining.

Hao Peng Hao Peng is currently an Assistant Professor with the School of Cyber Science and Technology, Beihang University. His current research interests include social text mining, representation learning, and deep learning.

Jianxin Li Jianxin Li is a professor in the School of Computer Science and Engineering, Beihang University, China. He received the Ph.D. degree from Beihang University in 2008. He was a visiting scholar in the Machine Learning Department of CMU in 2015, and a visting researcher of MSRA in 2011. His current research interests include big data, NLP, and industrial information security.

Lichao Sun Lichao Sun received the M.S. and B.S. degree from University of Nebraska Linoln, and the Ph.D. degree from University of Illinois at Chicago. He is an assistant professor at Lehigh University. His research interests include AI security, mobile security and privacy, etc.

Lingjuan Lyu Lingjuan Lyu received the Ph.D. degree in Department of Computer Science from The University of Melbourne. Before joining National University of Singapore, she was a research fellow in the Research School of Computer Science of Australian National University. Her research interests include distributed/federated machine/deep learning, machine learning fairness and system fairness.

Lihong Wang Lihong Wang, is currently a Professor in the National Computer Network Emergency Response Technical Team/Coordination Center of China. Her current research interests include information security, cloud computing, big data mining and analytics, information retrieval, and data mining.

Philip S. Yu Philip S. Yu received the B.S. Degree in E.E. from National Taiwan University, the M.S. and Ph.D. degrees in E.E. from Stanford University, and the M.B.A. degree from New York University. He is a Distinguished Professor in Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. Before joining UIC, Dr. Yu was with IBM, where he was manager of the Software Tools and Techniques department at the Watson Research Center. His research interest is on big data, including data mining, data stream, database and privacy. He has published more than 1000 papers in refereed journals and conferences.

Lifang He Lifang He received the B.S. degree in Computational Mathematics from Northwest Normal University, and the Ph.D. degree in Computer Science from South China University of Technology. Before joining Lehigh's faculty, she was a postdoctoral associate in the Department of Biostatistics, Epidemiology and Informatics within the Perelman School of Medicine at University of Pennsylvania, as well as the Weill Cornell Medical College of Cornell University. Her research interests include machine learning/deep learning, data mining, tensor analysis and biomedical informatics.