A Heterogeneous Propagation Graph Model for Rumor Detection under the Relationship among Multiple Propagation Subtrees

Guoyi Li[1,2], Jingyuan Hu[1,2], Yulei Wu[3], Xiaodan Zhang[1,2](✉), Wei Zhou[1,2], and Honglei Lyu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{liguoyi,hujingyuan,zhangxiaodan,zhouwei,lvhonglei}@iie.ac.cn
[3] Department of Computer Science, University of Exeter, Exeter, UK
Y.L.Wu@exeter.ac.uk

Abstract. Pervasive rumors in social networks have significantly harmed society due to their seditious and misleading effects. Existing rumor detection studies only consider practical features from a propagation tree, but ignore the important differences and potential relationships of subtrees under the same propagation tree. To address this limitation, we propose a novel heterogeneous propagation graph model to capture the relevance among different propagation subtrees, named Multi-subtree Heterogeneous Propagation Graph Attention Network (MHGAT). Specifically, we implicitly fuse potential relationships among propagation subtrees using the following three methods: 1) We leverage the structural logic of a tree to construct different types of propagation subtrees in order to distinguish the differences among multiple propagation subtrees; 2) We construct a heterogeneous propagation graph based on such differences, and design edge weights of the graph according to the similarity of propagation subtrees; 3) We design a propagation subtree interaction scheme to enhance local and global information exchange, and finally, get the high-level representation of rumors. Extensive experimental results on three real-world datasets show that our model outperforms the most advanced method.

Keywords: Rumor Detection · Heterogeneous Graph · Propagation Subtrees · Local and Global Relations · Message Passing

1 Introduction

Due to the popularity of Twitter, Facebook and other social media in recent years, a growing number of rumor generating methods have emerged. Taking the COVID-19 pandemic as an example, there were growing concerns about the spread of misinformation about the pandemic, known as the "information epidemic" [27]. Social media have been widely used to facilitate the spread of misinformation. These issues are even more pressing in that atmosphere since the information flowing through social media is directly related to human health and safety. It is therefore of paramount importance to effectively identify rumors.

Most existing efforts mainly focus on using linguistic features from text to detect rumors, ranging from deceptive clues to writing styles. For example, Li et al. [12] combined user information and text features to train an LSTM to capture their potential associations. Other algorithms such as the Bayesian network were applied to compute text-similarity of microblogs [11]. This kind of rumor detection methods was mainly to capture text features of rumors, which is vulnerable to the negative influence of forged text because the language used in social media is highly informal, ungrammatical, and dynamic.

To address the above issue, studies of rumor propagation structures have been carried out. For instance, Kumar et al. [10] proposed a new way to represent social-media conversations as propagation trees and used Tree LSTM models to capture conversation features. Ma et al. [19] proposed recursive neural models based on a bottom-up and a top-down tree-structured neural networks, to learn discriminative features from tweet's content by following their non-sequential propagation structures. Since temporal structural characteristics only concern the sequence of spreading rumors but ignore the consequence of rumor spreading, these approaches have significant limitations in terms of effectiveness. The structure of rumor dispersion also reflects important features of rumor spreading.

To consider such crucial features, researchers have started to apply graph convolution methods to detect rumors. Yu et al. [25] used GCN to realize the fusion of rumors in the propagation tree, the user information, and text features of retweets. Choi et al. [3] proposed a dynamic GCN to construct a time graph and utilized the characteristics of tweets published in adjacent times to strengthen the structural features of rumor propagation.

While the above methods have shown effectiveness of introducing the graph structure of data into a model, these approaches face two major shortcomings which make the rumor representation vulnerable to the local structural relationships and the characteristics of adjacent nodes. **First, existing studies only consider the aggregated information of each tweet and its neighbour, but ignore the important correlation of all retweets in the same propagation subtree. Second, the graphical structure of data ignores the potential impact among different propagation subtrees**.

To facilitate the understanding, Fig. 1 exemplifies the propagation structure of a (rumor) tweet "Says Bill O'Reilly wrote a post claiming that the coronavirus was created as a bioweapon by the Chinese government." In the first case, tweets x_1 and x_2 have the same characteristics ["article", "criminal acts"]; they have a certain correlation but no real connection. In fact, x_2 negates the basis of x_1. In the second case, x_1 and x_{11} incline that s is true and has a positive impact on the s. Even though x_{12} deems s was wrong and had more common characteristics with s, it can only affect s along the x_1, while x_1 and x_{11} prefer s is false and cannot well incorporate the features of the deeper retweets. The above two situations are common in rumor detection, and their cumulative impact may

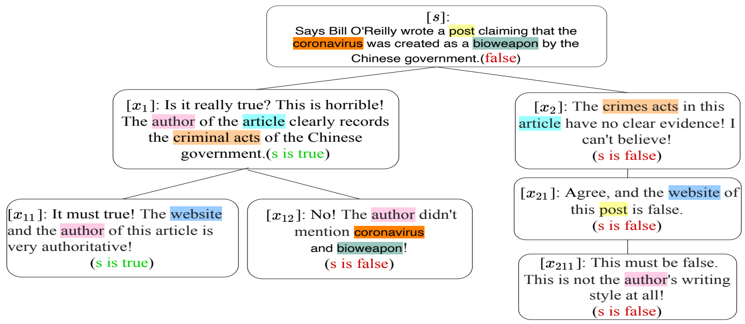


Fig. 1. An example of a false rumor.

lead to unexpected errors. Therefore, in this paper we propose to enhance the effect of rumor detection by constructing a local representation.

The starting point of our approach is an observation: tweets in a similar propagation location show certain relevance (such as $[x_1, x_{11}, x_{12}]$, $[x_2, x_{21}, x_{211}]$ in Fig. 1). Thus, we propose a new way of message passing to obtain the high-level representation of rumor propagation: (1) According to the structural logic of a tree and the spatial relationship among nodes, we model the different propagation subtrees of the tree where the nodes are located. We construct a heterogeneous propagation graph model with the weights of edges designed according to the propagation subtrees' similarity. (2) We initialize each node to integrate the relative temporal information carried by the parent tweet and the source post information, and apply structure-aware self-attention to propagation subtrees. (3) We design a two-layer attention mechanism to realize the interaction among propagation subtrees.

The main contribution of this paper can be summarized as follows:

- We propose a novel MHGAT model, which applies the propagation subtree as the computing unit to construct the heterogeneous propagation graph. It improves the performance of rumor detection by distinguishing the differences of local structures on the propagation tree.
- The model utilizes the heterogeneous propagation graph to guide the direction of message transmission. Moreover, the interaction between local information and global information is constructed to obtain the high-level rumor representation.
- The model fuses the parent tweet text features with the corresponding time information and the source text feature in appropriate places, to make the representation of the local structure more accurate.
- We conduct extensive experiments using three public real-world datasets. Experimental results show that our model significantly outperforms the state-of-the-art models in rumor classification and early detection tasks.

2 Related Work

Rumor detection aims to detect whether a tweet is a rumor according to the relevant information of the tweet published on the social media platform, such as text content and propagation mode.

Content-based Classification Methods : Content-based classification methods [7,24] generally detect rumors based on linguistic clues such as writing style [20], bag-of-words [4], temporal characteristics [17], etc. However, these methods relying only on the text content to detect rumors, ignore the correlation between tweets, and its accumulative effect on a large number of tweets can affect the performance of detection.

Propagation-based Classification Methods : Recent studies can be divided into two groups: Attention-based and GCN-based models. Attention-based models primarily utilize the attention mechanism to focus on pairs or sequences of posts with some inherent order [8,15,19,10]. Several recent works applied the transformer to enhance the representation learning for responsive tweets [8,15]. The difference lies in that Khoo et al. [8] defined time delay (the time interval between the tweet and retweet) as the intrinsic order, while Ma et al. [15] applied the topological order of the propagation tree as the inherent order. However, these methods are susceptible to the negative impact of unrelated tweets and require more time cost for detection. GCN-based models enhance the tweet representation by aggregating the features of related retweets [1,25]. For example, BiGCN [1] applied graph convolution to strengthen root features and learn local structure information. To better weight different types of neighbor nodes, in recent years several studies have applied heterogeneous graph model Graph Attention Network (GAT) that combines attention mechanism and GCN for rumour detection [26,13]. For example, Lin et al. [13] represented the propagation tree as an undirected interaction graph and utilized GAT aggregating information from parent and sibling nodes, taking the average representation as rumor representation that makes it difficult to distinguish the global structure of the rumor.

However, the above methods treat a tree's substructures as independent units, ignoring their differences and potential global associations. Our model will take advantage of the propagation tree structure and the heterogeneous graph model to construct the interaction between local and global information in order to enhance the representation of tweets in rumor detection.

3 Multi-subtree Heterogeneous propagation Graph Attention Network Model

This section details the proposed MHGAT algorithm as shown in Fig. 2. Our algorithm can be divided into four parts. First, we construct the heterogeneous propagation graph to refine different subtrees (substructures) of the ordinary

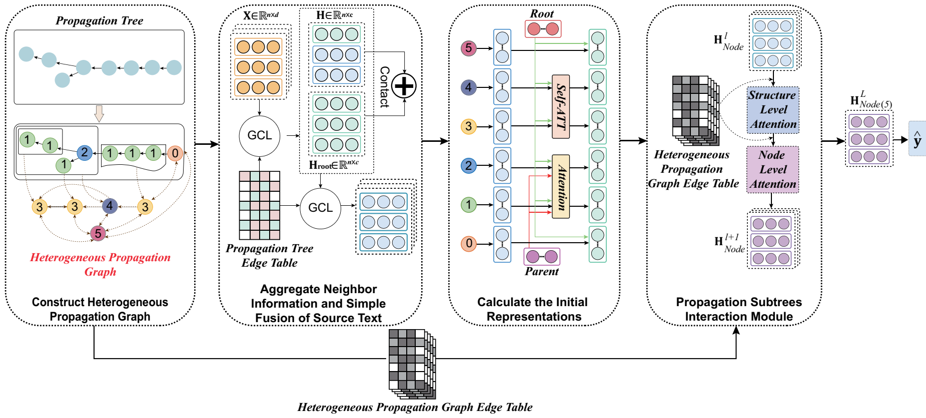


Fig. 2. Our proposed rumor detection model.

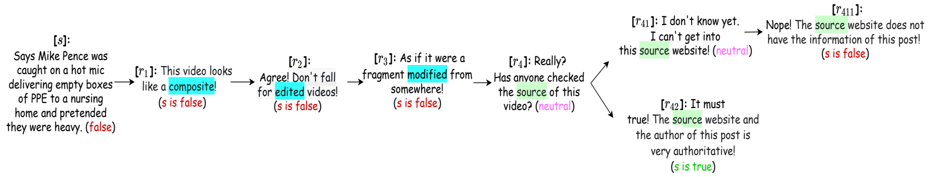


Fig. 3. Propagation tree of a false rumor.

propagation tree (Section 3.1). Second, to obtain a direct local representation, we utilize GCN to aggregate neighbor's features of the ordinary propagation tree (Section 3.2). Third, we get the initial representation of different subtrees in the heterogeneous propagation graph (Section 3.3). Finally, we design a heterogeneous graph convolution algorithm to realize the interaction between local and global information to enhance the rumor representation (Section 3.4).

Formally, let each node represent a tweet. The source node denotes the source tweet, and the children nodes are retweets that have responded to it directly. First, based on the retweet and the response relationships, we construct the origin event tree for a rumor c_i. In each training period, a propagation subtree has the probability p, of being discarded to reduce overfitting [1]. The probability of subtree pruning is positively correlated with the depth of the tree: $P_{drop} \propto dep(root\_of\_subtree)$. We denote the event tree after being discarded as $<V, E>$ (see Fig. 3).

3.1 Construct Heterogeneous Propagation Graph

The heterogeneous propagation graph $<V', E'>$ is designed to distinguish the differences of propagation subtrees better and address the two limitations mentioned above. This process is implemented with a general tree structure data processing method Depth-First-Search [21]. Our heterogeneous propagation graph

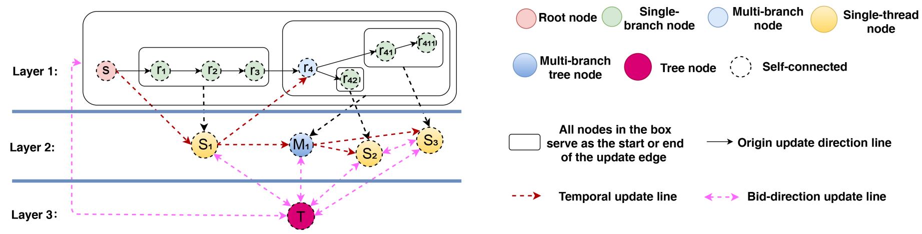


Fig. 4. The heterogeneous propagation graph is constructed by classifying the nodes of the original propagation tree in the first layer and building the nodes and edges of propagation subtrees in the last two layers. In addition, each node has a self-connected edge.

includes six types of structural nodes as shown in Fig. 4: (1) Root node ($Node_{(0)}$): Source tweet. (2) Single-branch node ($Node_{(1)}$): Leaf or the node with the single child except for the root node. (3) Multi-branch node ($Node_{(2)}$): The node with multiple branches except for the root node. (4) Single-thread node ($Node_{(3)}$): the node representing the single propagation thread without other branches. (5) Multi-branch tree node ($Node_{(4)}$): a propagation subtree with multiple branches. (6) Tree node ($Node_{(5)}$): a complete tree. Intuitively, we exemplify a false rumor claim and illustrate its propagation on twitter in Fig. 3. We observe that a group of tweets in the single-chain from r_1, $[r_1, r_2, r_3]$ tend to a point of view or a content, and construct their local representation S_1 in Fig. 3 to enhance features like ["composite", "edited", "modified"]. Moreover, we refer M_1 to the representation of $\{r_4, r_{41}, r_{411}, r_{42}\}$ which contains a stronger collection of different opinions about one content arising from the multi-branch node r_4. Essentially, multi-branch nodes have a broader direct impact influence than single-branch nodes. Finally, we refer to the tree node as the global representation to enhance rumor representation by realizing the interaction between the local information in the first five structure nodes and the global information.

In addition to the connection of nodes within the propagation subtree, we have added two effective connecting edges between $Node_{(3)}$ and $Node_{(4)}$:

(1) Considering that the nodes of a thread in the propagation tree have the corresponding time relationship (like $s \to r_1$ in Fig. 3), we now extend this feature to propagation subtrees in the heterogeneous propagation graph (such as $s \to S_1$ in Fig. 4). We define: when u and v are propagation subtrees of the type $Node_{(3)}$ or $Node_{(4)}$, r_i and r_j are retweet nodes in the propagation subtrees u and v respectively, where $i \neq j$, $u \neq v$. If r_i connects to r_j, u has a directed edge to v.

(2) Considering that two retweets forwarding the same tweet (the parent of r_i and r_j is the same node) may have similar characteristics (for example r_{42} and $\{r_{41}, r_{411}\}$ have common features ["source", "author"] in Fig. 3), and the propagation subtrees (u, v) are also related (S_2 and S_3 in Fig. 4), we define: when u and v are propagation subtrees of the type $Node_{(3)}$ or $Node_{(4)}$, r_i and

r_j are retweet nodes in propagation subtrees u and v respectively, where $i \neq j$. If Father(r_i) = Father(r_j), u has a undirected edge to v.

Normalization: Considering the large difference in the number of nodes of the same type connecting different nodes, it may have an adverse impact on model learning. We normalize the weights of the edges of the starting nodes of the same type. Among the neighbors pointing to node u, the node set of type i is marked as $N^i_{(u)}$, and the set size is marked as $\mathbf{Num}(N^i_{(u)})$. The edge regularization weight from any $v \in N^i_{(u)}$ to u is normalized to: $(\mathbf{Num}(N^i_{(u)}))^{-1}$. Thus, we get the normalized adjacency matrix $\tilde{\mathbf{A}}$ of heterogeneous propagation graph $<V', E'>$.

3.2 Aggregate neighbour information and simple fusion of root features

This module aims to strengthen the representation of nodes in the propagation tree $<V, E>$ by aggregating adjacent nodes and the source tweet. Graph convolution is an essential operation for aggregating neighbor information to extract local features. In addition, the source tweet can enhance the effect of rumor texts on retweets. As for nodes, let $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote the normalized adjacency matrix, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ represent the input signals of nodes of the propagation tree $<V, E>$. First, we aggregate neighbour's features from node embedding $\mathbf{X}$:

$$\mathbf{H} = \mathbf{ReLU}(\mathbf{AX}W_0). \tag{1}$$

Second, the aggregated features are fused with the root,

$$\mathbf{H}' = concat(\mathbf{H}, \mathbf{H_{root}}). \tag{2}$$

Last, we perform another layer of graph convolution to get a high-level representation of the node:

$$\tilde{\mathbf{X}} = \mathbf{ReLU}(\mathbf{AH}'W_1), \tag{3}$$

where $\mathbf{H}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times d}$ are the hidden feature matrices computed by the Graph Conventional Layer (GCL), $W_0 \in \mathbb{R}^{d \times c}, W_1 \in \mathbb{R}^{(c+c) \times d}$. W_0, W_1 are the filter parameter matrices of graph convolution layer, and $\mathbf{H_{root}}$ represents the root representation after first-layer graph convolution. $\tilde{\mathbf{X}}$ is the node representation after two layers of graph convolutional layers.

3.3 Calculate the Initial Representation

We apply the attention mechanism to fuse parent node and source text feature (root node) to enhance the representation of propagation subtrees in heterogeneous propagation graph $<V', E'>$, which can fuse the corresponding time and the source text information. For the root node ($Node_{(0)}$) and the tree node

$(Node_{(5)})$: the node is initialized to the representation of the processed root embedding: $\mathbf{H}_{(Node_{(0)})} = \mathbf{H}_{(Node_{(5)})} = \mathbf{X}_{root}$. For single-branch nodes and multi-branch nodes, we fuse the source text feature and the parent tweet text feature, and these embeddings are calculated as:

$$\mathbf{H}_{(Node_{(1)\sim(2)})} = \mathbf{ATTN}(\tilde{\mathbf{X}}_{(Node_{(1)\sim(2)})}, \mathbf{H}_{pr}), \tag{4}$$

where

$$\mathbf{H}_{pr} = concat(\mathbf{H}_{parent}, \mathbf{H}_{root}). \tag{5}$$

where **ATTN** is a function $f : \mathbf{X}_{key} \times \varphi \to \mathbf{X}_{val}$, which maps the feature vector $\mathbf{X}_{key}$ and candidate feature vector set φ to the weighted sum of elements in $\mathbf{X}_{val}$[22].

For the single-thread node $(Node_{(3)})$ and the multi-branch tree node $(Node_{(4)})$, these two types of nodes represent point sets, and we utilize attention mechanism to fuse the point sets into one representation:

$$\mathbf{H}_{(node_{(3)\sim(4)})} = \textbf{Self-ATT}(\tilde{\mathbf{X}}_{(Node_{(3)\sim(4)})}), \tag{6}$$

where **Self-ATT**(.) includes the fusion process of self-attention and attention fusion[22]. Moreover, the gated mechanism is applied to strengthen the root features to get a high-level representation:

$$\alpha = \sigma(W_r \mathbf{H}_{(node_{(3)\sim(4)})} + W_{root}\tilde{\mathbf{X}}_{root} + b), \tag{7}$$

$$\mathbf{H}_{(Node_{(3)\sim(4)})} = \alpha \times \mathbf{H}_{(node_{(3)\sim(4)})} + (1-\alpha)\tilde{\mathbf{X}}_{root}, \tag{8}$$

where $\sigma(\cdot)=\frac{1}{1+exp(\cdot)}$ is sigmoid activation function, and $W_r, W_{root} \in \mathbb{R}^{d\times 1}$, $b \in \mathbb{R}$ are parameters of the fusion gate.

Weight introduction: In addition to the regularized weights that eliminate quantitative differences, since these new potential links may introduce noise where not all neighbors are equal in contributing important information for the aggregation when modelling the propagation subtrees, we shall calculate the weight of links between propagation subtrees in heterogeneous propagation graph $<V', E'>$. To this end, we first use the cosine similarity $s(u, v) = h_u \cdot h_v^T/(|h_u| \cdot |h_v|)$ between nodes u and v to measure their similarity, where h is the embedding of the node. To properly define node's similarity, we introduce an asymmetric regularization term to balance the difference of the sum of similarity on every neighbor node:

$$\mathbf{R}_u(s(u,v)) = s(u,v)/\sum_{t}^{n} s(u,t), \tag{9}$$

where n is the set of u neighbor nodes. Combining the topology and attribute information, the similarity between u and v is

$$w(u,v) = W_{\tau_u\tau_v}(b(u,v) + \beta \cdot \mathbf{R}_u(h_u \cdot h_v^T / (|h_u| \cdot |h_v|)), \tag{10}$$

where β is a parameter to make a tradeoff between network topology and attributes, and $W_{\tau_u \tau_v}$ represents the trainable similarity relationship between propagation subtree type τ_u and τ_v. $b(u, v)$ is a network topology term: (1) If τ_u, $\tau_v \in [Node_{(0\sim2)}]$, $b(u, v) = 0$, which regards the points are the same in the topology. (2) If $\tau_u \in [Node_{(0\sim2)}]$, $\tau_v \in [Node_{(3\sim5)}]$, $b(u, v) = (-1)^{\delta(u,v)} \gamma_{\tau_u \tau_v}$, $\delta(u, v) = 1$ where γ is a trainable parameter, if u is a point in propagation subtree v, $\delta(u, v) = 0$. (3) If τ_u, $\tau_v \in [Node_{(3\sim4)}]$, $b(u, v) = n_u n_v / 2e$, n_u represents the number of points in propagation subtree u.

Therefore, for propagation subtrees, let $\tilde{\mathbf{A}}'$ represent the matrix $\tilde{\mathbf{A}}$ with weights introduced, and $\tilde{\mathbf{A}}'_s \in \mathbb{R}^{|n| \times |n_s|}$ denote the submatrix of $\tilde{\mathbf{A}}'$, whose rows represent all the nodes and columns denote their neighboring nodes with the type s.

3.4 Propagation Subtree Interaction Module

This module is designed to realize the interaction between local and global structral features in the heterogeneous propagation graph. In other words, tree nodes aggregate local structural information in each iteration while other structural nodes aggregate local and global structural information. It consists of two attention layers to aggregate various types of subtrees. First, we calculate the structure-level attention scores based on the node embedding h_u and the propagation subtree type embedding h_s:

$$\alpha_s = softmax(\mathbf{LeakyRelu}(w_s^T [h_u || h_s]), \tag{11}$$

$$s = \sum_{v' \in N_u} \tilde{\mathbf{A}}'_{uu'} h_{u'}, \tag{12}$$

where h_s is the sum of neighbouring node features, and $h_{u'}$ refers to the embedding of nodes $u' \in N_u$ with the same propagation subtree type s.

Then, as for the node-level attention part, given a specific node v with the structure type s and its neighboring node $u' \in N_u$ with the structure type s', we compute the node-level attention scores based on the node embeddings h_u and $h_{u'}$ with the structure-level attention weight α_s for the node u:

$$v_{uu'} = softmax(LeakyRelu(w_{node}^T \cdot \alpha_s [h_u || h_{u'}]), \tag{13}$$

where w_{node}^T is the attention vector. Then, we merge structure-level and node-level attention into heterogeneous propagation graph convolution.

$$\mathbf{H}^{(l+1)} = \sigma(\sum_{s \in Node_{(*)}} I_s \cdot \mathbf{H}_s^{(l)} \cdot W_s^{(l)}). \tag{14}$$

Here, I_s represents the attention matrix, whose element in the u^{th} row u'^{th} column is $v_{uu'}$.

Finally, after going through an L times propagation subtree interaction process, the label of the event $\tilde{S}$ is calculated as:

$$\tilde{y} = softmax(FC(\mathbf{H}^{L}_{Node_{(5)}})), \tag{15}$$

where $\tilde{y} \in \mathbb{R}^{1\times C}$ is a vector of probabilities for all the classes used to predict the label of the rumor.

4 Experiments

4.1 Datasets

Almost all prevalent datasets for experimental evaluation in the field of rumor detection come from two source platforms: Twitter and Sina Weibo. We evaluate the proposed model on three real-world datasets: *Twitter15*[18], *Twitter16*[18] and *Weibo*[14]. In all the three datasets, nodes refer to source tweets and retweets, edges represent response relationships, and features are the extracted top-5000 words in terms of the TF-IDF values. The Twitter15 and Twitter16 datasets contain four different labels, namely "false rumor" (FR), "non-rumor" (NR), "unverified" (UR), and "true rumor" (TR). Moreover, the Weibo dataset only contains binary labels, i.e., "true rumor" and "false rumor". Details of the three datasets are shown in Table 1.

4.2 Baselines and Evaluations Metrics

We compare our proposed model with the following baseline and state-of-the-art models. **ClaHi-GAT**[13]: An undirected interaction graph model utilizes GAT to capture interactions between posts with responsive parent-child or sibling relationships. **BiGCN**[1]: A bottom-up and a top-down tree-structured fusion model based on GCN for rumor detection. **PLAN**[8]: A transformer-based rumour detection model that can capture the interaction between any pair of tweets, even irrelevant ones. **RvNN**[19]: A bottom-up and a top-down tree-structured model based on recursive neural networks for rumor detection on Twitter. **SVM-TK**[18]: A SVM model uses Tree kernel to capture the propagation structure. **SVM-TS**[17]: A linear SVM classifier that uses content features to build a time-series model. **DTC**[2]: A decision tree-based model that utilizes a combination of news characteristics.

Table 1. Details of the datasets

Statistic	Twitter15	Twitter16	Weibo
# of source tweets	1490	818	4664
# of posts	331,612	204,820	3,805,656
# of users	276,663	173,487	2,746,818
# True rumors	374	205	2351
# False rumors	370	205	2313
# Unverified rumors	374	203	0
# Non-rumors	372	205	0

Table 2. Experimental results on *Weibo* dataset.

Metric	Class	DTC	SVM-TS	SVM-TK	RvNN	PLAN	BiGCN	ClaHi-GAT	MHGAT
Acc.	-	0.767	0.756	0.786	0.794	0.831	0.863	0.852	**0.914**
Prec.	F	0.735	0.732	0.916	0.833	0.823	0.971	0.953	**0.978**
	T	0.685	0.714	0.613	0.727	0.885	0.775	0.754	**0.841**
Rec.	F	0.763	0.804	0.819	0.783	0.841	0.717	0.739	**0.853**
	T	0.786	0.821	0.753	0.833	0.766	0.971	0.952	**0.985**
F_1	F	0.749	0.774	0.864	0.812	0.832	0.824	0.861	**0.868**
	T	0.732	0.717	0.773	0.808	0.821	0.862	0.842	**0.897**

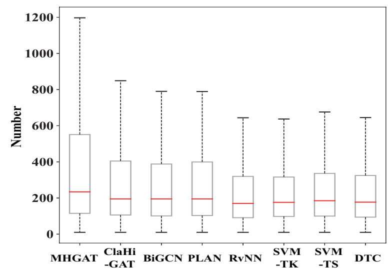
Table 3. Experimental results on *Twitter15* and *Twitter16*.

	Twitter15					*Twitter16*				
Method	Acc.	N	F	T	U	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1		F_1	F_1	F_1	F_1
DTC	0.625	0.716	0.519	0.642	0.523	0.607	0.652	0.432	0.573	0.739
SVM-TS	0.581	0.394	0.520	0.463	0.549	0.645	0.546	0.638	0.654	0.668
SVM-TK	0.705	0.619	0.756	0.485	0.835	0.732	0.814	0.713	0.745	0.801
RvNN	0.759	0.714	0.765	0.814	0.714	0.722	0.628	0.712	0.833	0.714
PLAN	0.795	0.784	0.810	0.793	0.802	0.825	0.846	0.803	0.774	0.832
BiGCN	0.814	0.772	0.827	0.830	0.786	0.816	0.751	0.839	0.904	0.781
ClaHi-GAT	0.823	0.805	0.843	0.894	0.807	0.838	0.763	0.864	0.892	0.816
MHGAT	**0.862**	**0.836**	**0.872**	**0.925**	**0.823**	**0.874**	**0.836**	**0.896**	**0.912**	**0.852**

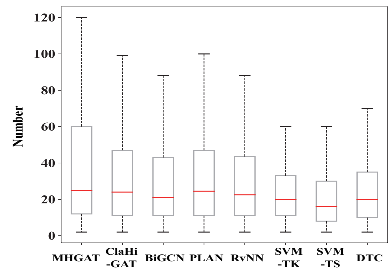
For a fair comparison, we adopt the same evaluation metrics that have already been widely used in existing work [5,6]. Thus, for the Weibo dataset, we evaluate the Accuracy (Acc.), Precision (Prec.), Recall (Rec.) and F_1 measure (F_1) on each class. For the two Twitter datasets, we evaluate the Accuracy (Acc.) and F_1 on each class.

4.3 Data Processing and Experiments Setup

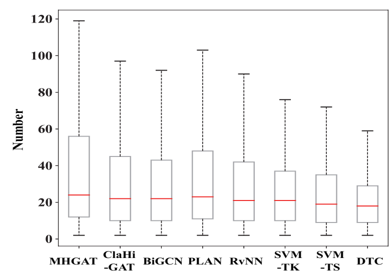
To be more realistic, we randomly select 15% of the instances as the development dataset that the model has not seen at all, and split the remaining instances into training and test datasets at a ratio of 4:1 in all datasets; this similar to the settings in existing studies [16,26]. In order to reduce the randomness, we repeat the experiments fifty times and take the average value as the result. We optimize the model using the Adam algorithm [9]. The dimension of each node's hidden feature vector is 128. The number of head K of self-attention is set to



(a) *Weibo* Dataset



(b) *Twitter15* Dataset



(c) *Twitter16* Dataset

Fig. 5. Comparison of the number of correctly detected rumor data (the vertical axis), where each rumor data is a propagation tree consisting of one source tweet and a number of retweets. The horizontal axis represents the proposed model and the various baselines.

8. The dropping rate in Subtree Drop is 0.1 for all three datasets. The training process is iterated upon 150 epochs and early stopping [23] is applied when the validation loss stops decreasing by 10 epochs.

4.4 Results and Analysis

Tables 2 and 3 show the performance of the proposed method and all comparison methods on Weibo and Twitter datasets. Compared with the content-based methods like DTC, SVM-TS, the propagation-based methods considering the propagation structure's characteristics, are generally more effective. The success rate of PLAN is higher than that of SVM-TK and RVNN that focus on propagation characteristics, because the potential relevance of all posts is considered in PLAN, but it tends to cause noises weakening the topological structure of the propagation tree. BiGCN and ClaHi-GAT pay more attention to the topology of propagation trees and aggregate the local characteristics of the propagation tree. The former demonstrates the effectiveness of incorporating the structure of dispersion and the source text features enhancement into rumor detection, while the latter shows the effectiveness of considering potentially associated tweets based on topological structures. However, these two methods can only take the average of all local representations as rumor representation, ignoring the differences among local representations and the impact of the global structure.

MHGAT considers the influence of the dispersion and the sequence structure of rumor propagation, the difference among local structures, and the interaction between local and global information. In addition, it strengthens the rumor representation by incorporating the source text feature and the parent text feature where appropriate. Thus, MHGAT outperforms all the baselines and state-of-the-art methods on all three datasets, especially in the large-scale *Weibo* dataset.

In order to further illustrate the detection performance of the model, we compare the number of correctly detected rumor data by different methods as shown in Fig. 5. By comparing box sizes and the upper and lower bounds, we found that methods (PLAN, BiGCN, ClaHi-GAT, MHGAT) that consider the local propagation structure and the potential correlation of posts tend to work better with most data than the other methods. Clearly, MHGAT has a wider upper and lower limit and can cover a broader range of data than the other methods. It proves that our method does not need a large amount of complex data to learn and can cope with the high-flow hot spot rumor, showing its outstanding performance in a more complex real-world scenario.

4.5 Ablation Study

To analyze the effect of each module of MHGAT, we conduct a series of ablation studies on different parts of the model. The ablation study is conducted in the following order: **w/o SBN**: Removing single-branch subtree nodes (SBN) and the related edges, and utilizing the remaining information on the graph for rumor detection. **w/o MBN**: Removing multi-branch subtree nodes (MBN) and the related edges, and utilizing the remaining information on the graph for rumor detection. **w/o STN**: Removing single-thread subtree nodes (STN) and the related edges, and utilizing the remaining information on the graph for rumor detection. **w/o MBTN**: Removing multi-branch subtree nodes (MBTN) and the related edges, and utilizing the remaining information on the graph for rumor detection. **w/o TN**: Removing the tree node (TN) and the related edges, and taking the mean representation of all nodes in the heterogeneous propagation graph as the final representation of the rumour for rumor detection.

We can observe the effect of removing all kinds of propagation subtrees covering local information in Table 4, which proves the universality of propagation subtrees and the necessity of classifying differences among local structures. Specifically, removing STN has the most significant impact on the results, and the accuracy on the Weibo, Twitter15 and Twitter16 datasets has dropped by 7.2%, 4.7% and 4.7%, respectively. This result is predictable. The information carried by the SBN is fragmented, whereas the information carried by the STN is able to cover the local relevance better and still has a better effect without SBN. Furthermore, there is a decrease in the accuracy rate without TN, but it is still higher than the baselines and the other variants of the ablation study due to the interaction among local subtrees in the interaction process, confirming the importance of local information interaction and the effect of the interaction

Table 4. The ablation study results on the *Weibo*, *Twitter15* and *Twitter16* datasets.

Models	*Weibo* Accuracy	*Twitter15* Accuracy	*Twitter16* Accuracy
MHGAT	0.914	0.862	0.874
w/o SBN	0.853	0.829	0.813
w/o MBN	0.871	0.837	0.849
w/o STN	0.842	0.815	0.827
w/o MBTN	0.883	0.841	0.845
w/o TN	0.889	0.847	0.853

between local and global information. Since the proposed method is integrated with the source text feature and the parent text feature, it is necessary to analyze the effects of each component. As shown in Fig. 6, we compare the results of the complete model and its variants and find that the complete model is better than the ones without the fusion of source text feature or parent text feature. This shows incorporating the source text feature and the corresponding time information of the parent node in appropriate places can improve the performance of our model.

Moreover, when the model introduces implicit links between subtrees, not all neighbors can contribute important information to the aggregation. Thus we introduce weights for subtree aggregation. As shown in Fig. 6, the model with added weight is better than the model without weight, which proves that the weight we designed reasonably solves the noise problem introduced by implicit links and further enhances the effect of our rumor detection model.

4.6 Early Detection

One of the most crucial tasks in rumor detection is the early detection of rumors. In the early rumor detection task, we compare different detection methods at elapsed time checkpoints. As shown in Fig. 7, from the performance of our method and the baseline method on different time delays in the Twitter and

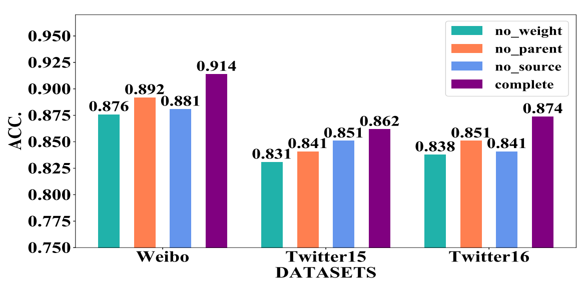


Fig. 6. Comparison of MHGAT and its variants.

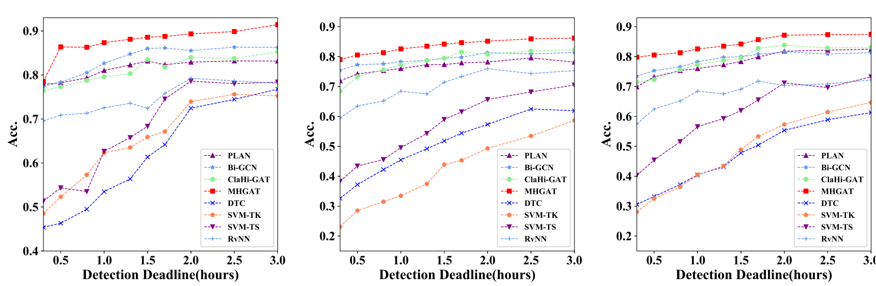
Weibo datasets, it can be seen that our method achieves higher accuracy very quick as soon as the initial broadcast of the source and can still maintain higher accuracy as the time delay goes up. It is worth noting that some baselines decrease slightly when the time delay increases. This is because as the rumor is propagated, more similar structural and semantic information shows, and more noises are introduced simultaneously. The results show that our model is more suitable for a complex real-world case and has a better stability.

5 Conclusions

This paper proposed a novel Multi-subtree Heterogeneous Propagation Graph Attention Network, which is used for social media rumor detection. This method refined propagation subtrees of the rumor propagation tree, strengthened the interaction between local and global structure information, and improved the ability to learn high-level rumor representation, hence achieving the best performance. Extensive experiments proved the superiority of the proposed method. However, one of the existing obstacles of rumor detection is the performance degradation caused by data uncertainty. To address this issue, in future we will study how to use uncertainty estimation to explain the model's performance in rumor propagation.

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: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 549–556 (2020)
2. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 675–684 (2011)



(a) *Weibo* Dataset (b) *Twitter15* Dataset (c) *Twitter16* Dataset

Fig. 7. Results of early rumor detection.

3. Choi, J., Ko, T., Choi, Y., Byun, H., Kim, C.k.: Dynamic graph convolutional networks with attention mechanism for rumor detection on social media. Plos one **16**(8), e0256039 (2021)
4. Enayet, O., El-Beltagy, S.R.: Niletmrg at semeval-2017 task 8: Determining rumour and veracity support for rumours on twitter. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017). pp. 470–474 (2017)
5. Fuller, C.M., Biros, D.P., Wilson, R.L.: Decision support for determining veracity via linguistic-based cues. Decision Support Systems **46**(3), 695–703 (2009)
6. Giudice, K.D.: Crowdsourcing credibility: The impact of audience feedback on web page credibility. Proceedings of the American Society for Information Science and Technology **47**(1), 1–9 (2010)
7. Jin, Z., Cao, J., Zhang, Y., Luo, J.: News verification by exploiting conflicting social viewpoints in microblogs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 30 (2016)
8. Khoo, L.M.S., Chieu, H.L., Qian, Z., Jiang, J.: Interpretable rumor detection in microblogs by attending to user interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8783–8790 (2020)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Kumar, S., Carley, K.M.: Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5047–5058 (2019)
11. Li, C., Liu, F., Li, P.: Text similarity computation model for identifying rumor based on bayesian network in microblog. Int. Arab J. Inf. Technol. **17**(5), 731–741 (2020)
12. Li, Q., Zhang, Q., Si, L.: Rumor detection by exploiting user credibility information, attention and multi-task learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1173–1179 (2019)
13. Lin, H., Ma, J., Cheng, M., Yang, Z., Chen, L., Chen, G.: Rumor detection on twitter with claim-guided hierarchical graph attention networks. arXiv preprint arXiv:2110.04522 (2021)
14. Liu, X., Nourbakhsh, A., Li, Q., Fang, R., Shah, S.: Real-time rumor debunking on twitter. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 1867–1870 (2015)
15. Ma, J., Gao, W.: Debunking rumors on twitter with tree transformer. ACL (2020)
16. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks (2016)
17. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 1751–1754 (2015)
18. Ma, J., Gao, W., Wong, K.F.: Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics (2017)
19. Ma, J., Gao, W., Wong, K.F.: Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics (2018)
20. Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., Stein, B.: A stylometric inquiry into hyperpartisan and fake news. arXiv preprint arXiv:1702.05638 (2017)
21. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM journal on computing **1**(2), 146–160 (1972)

22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
23. Yao, Y., Rosasco, L., Caponnetto, A.: On early stopping in gradient descent learning. Constructive Approximation **26**(2), 289–315 (2007)
24. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T., et al.: A convolutional approach for misinformation identification. In: IJCAI. pp. 3901–3907 (2017)
25. Yu, K., Jiang, H., Li, T., Han, S., Wu, X.: Data fusion oriented graph convolution network model for rumor detection. IEEE Transactions on Network and Service Management **17**(4), 2171–2181 (2020)
26. Yuan, C., Ma, Q., Zhou, W., Han, J., Hu, S.: Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In: 2019 IEEE International Conference on Data Mining (ICDM). pp. 796–805. IEEE (2019)
27. Zarocostas, J.: How to fight an infodemic. The Lancet **395**(10225), 676 (2020)