Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks

Tian Bian,1,2 Xi Xiao,1 Tingyang Xu,2 Peilin Zhao,2 Wenbing Huang,2 Yu Rong,2 Junzhou Huang2
1Tsinghua University
2Tencent AI Lab
bt18@mails.tsinghua.edu.cn, xiaox@sz.tsinghua.edu.cn, hwenbing@126.com, yu.rong@hotmail.com
{tingyangxu, masonzhao, joehhuang}@tencent.com

Abstract
Social media has been developing rapidly in public due to its nature of spreading new information, which leads to rumors being circulated. Meanwhile, detecting rumors from such massive information in social media is becoming an arduous challenge. Therefore, some deep learning methods are applied to discover rumors through the way they spread, such as Recursive Neural Network (RvNN) and so on. However, these deep learning methods only take into account the patterns of deep propagation but ignore the structures of wide dispersion in rumor detection. Actually, propagation and dispersion are two crucial characteristics of rumors. In this paper, we propose a novel bi-directional graph model, named Bi-Directional Graph Convolutional Networks (Bi-GCN), to explore both characteristics by operating on both top-down and bottom-up propagation of rumors. It leverages a GCN with a top-down directed graph of rumor spreading to learn the patterns of rumor propagation; and a GCN with an opposite directed graph of rumor diffusion to capture the structures of rumor dispersion. Moreover, the information from source post is involved in each layer of GCN to enhance the influences from the roots of rumors. Encouraging empirical results on several benchmarks confirm the superiority of the proposed method over the state-of-the-art approaches.

Introduction
With the rapid development of the Internet, social media has become a convenient online platform for users to obtain information, express opinions and communicate with each other. As more and more people are keen to participate in discussions about hot topics and exchange their opinions on social media, many rumors appear. Due to a large number of users and easy access to social media, rumors can spread widely and quickly on social media, bringing huge harm to society and causing a lot of economic losses. Therefore, regarding to the potential panic and threat caused by rumors, it is urgent to come up with a method to identify rumors on social media efficiently and as early as possible.

Conventional detection methods mainly adopt handcrafted features such as user characteristics, text contents and propagation patterns to train supervised classifiers, e.g., Decision Tree (Castillo, Mendoza, and Poblete 2011), Random Forest (Kwon et al. 2013), Support Vector Machine (SVM) (Yang et al. 2012). Some studies apply more effective features, such as user comments (Giudice 2010), temporal-structural features (Wu, Yang, and Zhu 2015), and the emotional attitude of posts (Liu et al. 2015). However, those methods mainly rely on feature engineering, which is very time-consuming and labor-intensive. Moreover, those handcrafted features are usually lack of high-level representations extracted from the propagation and the dispersion of rumors.

Recent studies have exploited deep learning methods that mine high-level representations from propagation path/trees or networks to identify rumors. Many deep learning models such as Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Recursive Neural Networks (RvNN) (Ma et al. 2016; Ma, Gao, and Wong 2018) are employed since they are capable to learn sequential features from rumor propagation along time. However, these approaches have a significant limitation on efficiency since temporal-structural features only pay attention to the sequential propagation of rumors but neglect the influences of rumor dispersion. The structures of rumor dispersion also indicate some spreading behaviors of rumors. Thus, some studies have tried to involve the information from the structures of rumor dispersion by invoking Convolutional Neural Network (CNN) based methods (Yu et al. 2017; 2019). CNN-based methods can obtain the correlation features within local neighbors but cannot handle the global structural relationships in graphs or trees (Bruna et al. 2014). Therefore, the global structural features of rumor dispersion are ignored in these approaches. Actually, CNN is not designed to learn high-level representations from structured data but Graph Convolutional Network (GCN) is (Kipf and Welling 2017).

So can we simply apply GCN to rumor detection since it has successfully made progress in various fields, such as social networks (Hamilton, Ying, and Leskovec 2017), physical systems (Battaglia et al. 2016), and chemical drug discovery (Defferrard, Bresson, and Vandergheynst 2016)? The answer is no. As shown in Figure 1(a), GCN, or called undirected GCN (UD-GCN), only aggregates information relied on the relationships among relevant posts but loses the se-
In recent years, automatic rumor detection on social media has attracted a lot of attention. Most previous work for rumor detection mainly focuses on extracting rumor features from the text contents, user profiles and propagation structures to learn a classifier from labeled data (Castillo, Menendez, and Poblete 2011; Yang et al. 2012; Kwon et al. 2013; Liu et al. 2015; Zhao, Resnick, and Mei 2015). Ma et al. (Ma et al. 2015) classified the rumor by using the time-series to model the variation of handcrafted social context features. Wu et al. (Wu, Yang, and Zhu 2015) proposed a graph kernel-based hybrid SVM classifier by combining the RBF kernel with a random-walk-based graph kernel. Ma et al. (Ma, Gao, and Wong 2017) constructed a propagation tree kernel to detect rumors by evaluating the similarities between their propagation tree structures. These methods not only were ineffective but also heavily relied on handcrafted feature engineering to extract informative feature sets.

In order to automatically learn high-level features, a number of recent methods were proposed to detect rumor based on deep learning models. Ma et al. utilized Recurrent Neural Networks (RNN) to capture the hidden representation from temporal content features (Ma et al. 2016). Chen et al. (Chen et al. 2018) improved this approach by combining attention mechanisms with RNN to focus on text features with different attention, which are crucial to identify rumors in real time and prevent them from spreading, Bi-GCN also achieves much higher effectiveness.

**Related Work**

In recent years, automatic rumor detection on social media has attracted a lot of attention. Most previous work for rumor detection mainly focuses on extracting rumor features from the text contents, user profiles and propagation structures to learn a classifier from labeled data (Castillo, Menendez, and Poblete 2011; Yang et al. 2012; Kwon et al. 2013; Liu et al. 2015; Zhao, Resnick, and Mei 2015). Ma et al. (Ma et al. 2015) classified the rumor by using the time-series to model the variation of handcrafted social context features. Wu et al. (Wu, Yang, and Zhu 2015) proposed a graph kernel-based hybrid SVM classifier by combining the RBF kernel with a random-walk-based graph kernel. Ma et al. (Ma, Gao, and Wong 2017) constructed a propagation tree kernel to detect rumors by evaluating the similarities between their propagation tree structures. These methods not only were ineffective but also heavily relied on handcrafted feature engineering to extract informative feature sets.

In order to automatically learn high-level features, a number of recent methods were proposed to detect rumor based on deep learning models. Ma et al. utilized Recurrent Neural Networks (RNN) to capture the hidden representation from temporal content features (Ma et al. 2016). Chen et al. (Chen et al. 2018) improved this approach by combining attention mechanisms with RNN to focus on text features with different attention, which are crucial to identify rumors in real time and prevent them from spreading, Bi-GCN also achieves much higher effectiveness.

**Related Work**

In recent years, automatic rumor detection on social media has attracted a lot of attention. Most previous work for rumor detection mainly focuses on extracting rumor features from the text contents, user profiles and propagation structures to learn a classifier from labeled data (Castillo, Menendez, and Poblete 2011; Yang et al. 2012; Kwon et al. 2013; Liu et al. 2015; Zhao, Resnick, and Mei 2015). Ma et al. (Ma et al. 2015) classified the rumor by using the time-series to model the variation of handcrafted social context features. Wu et al. (Wu, Yang, and Zhu 2015) proposed a graph kernel-based hybrid SVM classifier by combining the RBF kernel with a random-walk-based graph kernel. Ma et al. (Ma, Gao, and Wong 2017) constructed a propagation tree kernel to detect rumors by evaluating the similarities between their propagation tree structures. These methods not only were ineffective but also heavily relied on handcrafted feature engineering to extract informative feature sets.
as a special message-passing model for either undirected graphs or directed graphs. Later on, Bruna et al. (Bruna et al. 2014) theoretically analyzed graph convolutional methods for undirected graphs based on the spectral graph theory. Subsequently, Defferrard et al. (Defferrard, Bresson, and Vandergheynst 2016) developed a method named the Chebyshev Spectral CNN (ChebNet) and used the Chebyshev polynomials as the filter. After this work, Kipf et al. (Kipf and Welling 2017) presented a first-order approximation of ChebNet (1stChebNet), where the information of each node is aggregated from the node itself and its neighboring nodes. Our rumor detection model is inspired by the GCN.

Preliminaries

We introduce some fundamental concepts that are necessary for our method. First the notation used in this paper is as follows.

**Notation**

Let \( C = \{c_1, c_2, ..., c_m\} \) be the rumor detection dataset, where \( c_i \) is the \( i \)-th event and \( m \) is the number of events. \( c_i = \{r_i, w_{i1}, w_{i2}, ..., w_{in_i - 1}; G_i\} \), where \( n_i \) refers to the number of posts in \( c_i \), \( r_i \) is the source post, each \( w_{ij} \) represents the \( j \)-th relevant responsive post, and \( G_i \) refers to the propagation structure. Specifically, \( G_i \) is defined as a graph \( \langle V_i, E_i \rangle \) with \( r_i \) being the root node (Wu, Yang, and Zhu 2015; Ma, Gao, and Wong 2017), where \( V_i = \{r_i, w_{i1}, ..., w_{in_i - 1}\} \), and \( E_i = \{e_{it}|s, t = 0, \ldots, n_i - 1\} \) that represents the set of edges from responded posts to the retweeted posts or responsive posts, as shown in Figure 1(b). For example, if \( w_{i2} \) has a response to \( w_{i1} \), there will be an directed edge \( w_{i1} \rightarrow w_{i2} \), i.e., \( e_{12} \). If \( w_{i1} \) has a response to \( r_i \), there will be an directed edge \( r_i \rightarrow w_{i1} \), i.e., \( e_{01} \). Denote \( A_i \in \{0, 1\}^{n_i \times n_i} \) as an adjacency matrix where

\[
a_{ts} = \begin{cases} 1, & \text{if } e_{st} \in E_i \\ 0, & \text{otherwise} \end{cases}
\]

Denote \( X_i = [x_{i0}^T, x_{i1}^T, ..., x_{in_i - 1}]^T \) as a feature matrix extracted from the posts in \( c_i \), where \( x_{i0} \) represents the feature vector of \( r_i \), and each other row feature \( x_{ij} \) represents the feature vector of \( w_{ij} \).

Moreover, each event \( c_i \) is associated with a ground-truth label \( y_i \in \{F, T\} \) (i.e., False Rumor or True Rumor). In some cases, the label \( y_i \) is one of the four finer-grained classes \( \{N, F, T, U\} \) (i.e., Non-rumor, False Rumor, True Rumor, and Unverified Rumor) (Ma, Gao, and Wong 2017; Zubiaga et al. 2018). Given the dataset, the goal of rumor detection is to learn a classifier

\[
f : C \rightarrow Y,
\]

where \( C \) and \( Y \) are the sets of events and labels respectively, to predict the label of an event based on text contents, user information and propagation structure constructed by the related posts from that event.

**Graph Convolutional Networks**

Recently, there is an increasing interest in generalizing convolutions to the graph domain. Among all the existing works, GCN is one of the most effective convolution models, whose convolution operation is considered as a general “message-passing” architecture as follows:

\[
H_k = M(\mathbf{A}, H_{k-1}; W_{k-1}),
\]

(1)

where \( H_k \in \mathbb{R}^{n \times rk} \) is the hidden feature matrix computed by the \( k-th \) Graph Convolutional Layer (GCL) and \( M \) is the message propagation function, which depends on the adjacency matrix \( \mathbf{A} \), the hidden feature matrix \( H_{k-1} \) and the trainable parameters \( W_{k-1} \).

There are many kinds of message propagation functions \( M \) for GCN (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016). Among them, the message propagation function defined in the first-order approximation of ChebNet (1stChebNet) (Kipf and Welling 2017) is as follows:

\[
H_k = M(\mathbf{A}, H_{k-1}; W_{k-1}) = \sigma(\mathbf{A}H_{k-1}W_{k-1}).
\]

(2)

In the above equation \( \hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \) is the normalized adjacency matrix, where \( \hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N \) (i.e., adding self-connection), \( \mathbf{D} = \sum_{j} \hat{\mathbf{A}}_{ij} \) that represents the degree of the \( i-th \) node; \( W_{k-1} \in \mathbb{R}^{rk \times nk} \); and \( \sigma(\cdot) \) is an activation function, e.g., the ReLU function.

**DropEdge**

DropEdge is a novel method to reduce over-fitting for GCN-based models (Rong et al. 2019). In each training epoch, it randomly drops out edges from the input graphs to generate different deformed copies with certain rate. As a result, this method augments the randomness and the diversity of input data, just like rotating or flipping images at random.

Formally, suppose the total number of edges in the graph \( \mathbf{A} \) is \( N_e \) and the dropping rate is \( p \), then the adjacency matrix after DropEdge, \( \mathbf{A}' \), is computed as below:

\[
\mathbf{A}' = \mathbf{A} - \mathbf{A}_{drop}
\]

(3)

where \( \mathbf{A}_{drop} \) is the matrix constructed using \( N_e \times p \) edges randomly sampled from the original edge set.

**Bi-GCN Rumor Detection Model**

In this section, we propose an effective GCN-based method for rumor detection based on the rumor propagation and the rumor dispersion, named as Bi-directional Graph Convolutional Networks (Bi-GCN). The core idea of Bi-GCN is to learn suitable high-level representations from both rumor propagation and rumor dispersion. In our Bi-GCN model, two-layer 1stChebNet are adopted as the fundamental GCN components. As shown in Figure 2, we elaborate the rumor detection process using Bi-GCN in 4 steps.

We first discuss how to apply the Bi-GCN model to one event, i.e., \( c_i \rightarrow y_i \) for the \( i \)-th event. The other events are calculated in the same manner. To better present our method, we omit the subscript \( i \) in the following content.
Figure 2: Our Bi-GCN rumor detection model. \( \mathbf{X} \) denotes the original feature matrix input to the Bi-GCN model, and \( \mathbf{H} \) is the hidden features matrix generated from the \( k \)−th GCL. \( \mathbf{X}^{\text{root}} \) and \( \mathbf{H}^{\text{root}} \) represents the matrix extended by the features of source post.

1 Construct Propagation and Dispersion Graphs

Based on the retweet and response relationships, we construct the propagate structure \( \langle V, E \rangle \) for a rumor event \( c_i \). Then, let \( \mathbf{A} \in \mathbb{R}^{n_i \times n_i} \) and \( \mathbf{X} \) be its corresponding adjacency matrix and feature matrix of \( c_i \) based on the spreading tree of rumors, respectively. \( \mathbf{A} \) only contains the edges from the upper nodes to the lower nodes as illustrated in Figure 1(b). At each training epoch, \( p \) percentage of edges are dropped via Eq. (3) to form \( \mathbf{A}' \), which avoid penitential over-fitting issues (Rong et al. 2019). Based on \( \mathbf{A}' \) and \( \mathbf{X} \), we can build our Bi-GCN model. Our Bi-GCN consists of two components: a Top-Down Graph Convolutional Network (TD-GCN) and a Bottom-Up Graph Convolutional Network (BU-GCN). The adjacency matrices of two components are different. For TD-GCN, the adjacency matrix is represented as \( \mathbf{A}^{TD} = \mathbf{A}' \). Meanwhile, for BU-GCN, the adjacency matrix is \( \mathbf{A}^{BU} = \mathbf{A}'^T \). TD-GCN and BU-GCN adopt the same feature matrix \( \mathbf{X} \).

2 Calculate the High-level Node Representations

After the DropEdge operation, the top-down propagation features and the bottom-up propagation features are obtained by TD-GCN and BU-GCN, respectively. By substituting \( \mathbf{A}^{TD} \) and \( \mathbf{X} \) to Eq. (2) over two layers, we write the equations for TD-GCN as below:

\[
\mathbf{H}_1^{TD} = \sigma \left( \mathbf{A}^{TD} \mathbf{X} \mathbf{W}_0^{TD} \right),
\]

\[
\mathbf{H}_2^{TD} = \sigma \left( \mathbf{A}^{TD} \mathbf{H}_1^{TD} \mathbf{W}_1^{TD} \right),
\]

where \( \mathbf{H}_1^{TD} \in \mathbb{R}^{n \times v_1} \) and \( \mathbf{H}_2^{TD} \in \mathbb{R}^{n \times v_2} \) represent the hidden features of two layer TD-GCN. \( \mathbf{W}_0^{TD} \in \mathbb{R}^{d \times v_1} \) and \( \mathbf{W}_1^{TD} \in \mathbb{R}^{v_1 \times v_2} \) are the filter parameter matrices of TD-GCN. Here we adopt ReLU function as the activation function, \( \sigma(\cdot) \). Dropout (Srivastava et al. 2014) is applied on GCN Layers (GCLs) to avoid over-fitting. Similar to Eqs. (4) and (5), we calculate the bottom-up hidden features \( \mathbf{H}_1^{BU} \) and \( \mathbf{H}_2^{BU} \) for BU-GCN in the same manner as Eq. (4) and Eq. (5).

3 Root Feature Enhancement

As we know, the source post of a rumor event always has abundant information to make a wide impact. It is necessary to better make use of the information from the source post, and learn more accurate node representations from the relationship between nodes and the source post.

Consequently, besides the hidden features from TD-GCN and BU-GCN, we propose an operation of root feature enhancement to improve the performance of rumor detection as shown in Figure 2. Specifically, for TD-GCN at the \( k \)-th GCL, we concatenate the hidden feature vectors of every nodes with the hidden feature vector of the root node from the \( (k-1) \)-th GCL to construct a new feature matrix as

\[
\mathbf{H}_k^{TD} = \text{concat}(\mathbf{H}_k^{TD}, (\mathbf{H}_{k-1}^{TD})^{\text{root}})
\]

with \( \mathbf{H}_0^{TD} = \mathbf{X} \). Therefore, we express TD-GCN with the root feature enhancement by replacing \( \mathbf{H}_1^{TD} \) in Eq. (5) with \( \mathbf{H}_1^{TD} = \text{concat}(\mathbf{H}_1^{TD}, \mathbf{X}^{\text{root}}) \), and then get \( \mathbf{H}_2^{TD} \) as follows:

\[
\mathbf{H}_2^{TD} = \sigma \left( \mathbf{A}^{TD} \mathbf{H}_1^{TD} \mathbf{W}_1^{TD} \right),
\]

\[
\mathbf{H}_2^{TD} = \text{concat}(\mathbf{H}_2^{TD}, (\mathbf{H}_1^{TD})^{\text{root}}).
\]
Similarly, the hidden feature metrics of BU-GCN with root feature enhancement, \( \tilde{H}_1^{BU} \) and \( \tilde{H}_2^{BU} \), are obtained in the same manner as Eq. (7) and Eq. (8).

### 4 Representations of Propagation and Dispersion for Rumor Classification

The representations of propagation and dispersion are the aggregations from the node representations of TD-GCN and BU-GCN, respectively. Here we employ mean-pooling operators to aggregate information from these two sets of the node representations. It is formulated as

\[
S_{TD}^{BU} = \text{MEAN}(H_2^{TD}), \quad (9)
\]

\[
S_{BU}^{BU} = \text{MEAN}(H_2^{BU}). \quad (10)
\]

Then, we concatenate the representations of propagation and the representation of dispersion to merge the information as

\[
S = \text{concat}(S_{TD}, S_{BU}). \quad (11)
\]

Finally, the label of the event \( \hat{y} \) is calculated via several full connection layers and a softmax layer:

\[
\hat{y} = \text{Softmax}(FC(S)). \quad (12)
\]

where \( \hat{y} \in \mathbb{R}^{1 \times C} \) is a vector of probabilities for all the classes used to predict the label of the event.

We train all the parameters in the Bi-GCN model by minimizing the cross-entropy of the predictions and ground truth distributions, \( Y \), over all events, \( C \). \( L_2 \) regularizer is applied in the loss function over all the model parameters.

### Experiments

In this section, we first evaluate the empirical performance of our proposed Bi-GCN method in comparison with several baseline models. Then, we investigate the effect of each variant of the proposed method. Finally, we also examine the capability of early rumor detection for both the proposed method and the compared methods.

### Settings and Datasets

#### Datasets

We evaluate our proposed method on three real-world datasets: Weibo (Ma et al. 2016), Twitter15 (Ma, Gao, and Wong 2017), and Twitter16 (Ma, Gao, and Wong 2017). Weibo and Twitter are the most popular social media sites in China and the U.S., respectively. In all the three datasets, nodes refer to users, edges represent retweet or response relationships, and features are the extracted top-5000 words in terms of the TF-IDF values as mentioned in the Bi-GCN Rumor Detection Model Section. The Weibo dataset contains two binary labels: False Rumor (F) and True Rumor (T), while Twitter15 and Twitter16 datasets contain four labels: Non-rumor (N), False Rumor (F), True Rumor (T), and Unverified Rumor (U). The label of each event in Weibo is annotated according to Sina community management center, which reports various misinformation (Ma et al. 2016). And the label of each event in Twitter15 and Twitter16 is annotated according to the veracity tag of the article in rumor debunking websites (e.g., snopes.com, Emergent.info, etc) (Ma, Gao, and Wong 2017). The statistics of the three datasets are shown in Table 1.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Weibo</th>
<th>Twitter15</th>
<th>Twitter16</th>
</tr>
</thead>
<tbody>
<tr>
<td># of posts</td>
<td>3,805,656</td>
<td>331,612</td>
<td>204,820</td>
</tr>
<tr>
<td># of Users</td>
<td>2,746,818</td>
<td>276,663</td>
<td>173,487</td>
</tr>
<tr>
<td># of events</td>
<td>4664</td>
<td>1490</td>
<td>818</td>
</tr>
<tr>
<td># of True rumors</td>
<td>2351</td>
<td>374</td>
<td>205</td>
</tr>
<tr>
<td># of False rumors</td>
<td>2313</td>
<td>370</td>
<td>205</td>
</tr>
<tr>
<td># of Unverified rumors</td>
<td>0</td>
<td>374</td>
<td>203</td>
</tr>
<tr>
<td># of Non-rumors</td>
<td>0</td>
<td>372</td>
<td>205</td>
</tr>
<tr>
<td>Avg. time length / event</td>
<td>2,460.7 Hours</td>
<td>1,337 Hours</td>
<td>848 Hours</td>
</tr>
<tr>
<td>Avg. # of posts / event</td>
<td>816</td>
<td>223</td>
<td>251</td>
</tr>
<tr>
<td>Max # of posts / event</td>
<td>59,318</td>
<td>1,768</td>
<td>2,765</td>
</tr>
<tr>
<td>Min # of posts / event</td>
<td>10</td>
<td>55</td>
<td>81</td>
</tr>
</tbody>
</table>

#### Experimental Setup

We compare the proposed method with some state-of-the-art baselines, including:

- **DTC** (Castillo, Mendoza, and Poblete 2011): A rumor detection method using a Decision Tree classifier based on various handcrafted features to obtain information credibility.
- **SVM-RBF** (Yang et al. 2012): A SVM-based model with RBF kernel, using handcrafted features based on the overall statistics of the posts.
- **SVM-TS** (Ma et al. 2015): A linear SVM classifier that leverages handcrafted features to construct time-series model.
- **SVM-TK** (Ma, Gao, and Wong 2017): A SVM classifier with a propagation Tree Kernel on the basis of the propagation structures of rumors.
- **RvNN** (Ma, Gao, and Wong 2018): A rumor detection approach based on tree-structured recursive neural networks with GRU units that learn rumor representations via the propagation structure.
- **PPC,RNN+CNN** (Liu and Wu 2018): A rumor detection model combining RNN and CNN, which learns the rumor representations through the characteristics of users in the rumor propagation path.
- **Bi-GCN**: Our GCN-based rumor detection model utilizing the Bi-directional propagation structure.

We implement DTC and SVM-based models with scikit-learn\(^1\); PPC,RNN+CNN with Keras\(^2\); RvNN and our method with Pytorch\(^3\). To make a fair comparison, we randomly split the datasets into five parts, and conduct 5-fold cross-validation to obtain robust results. For the Weibo dataset, we evaluate the Accuracy (Acc.) over the two categories and Precision (Prec.), Recall (Rec.), F1 measure (\(F_1 \)) on each class. For the two Twitter datasets, we evaluate Acc. over the four categories and \(F_1 \) on each class. The

\(^{1}\)https://scikit-learn.org

\(^{2}\)https://keras.io/

\(^{3}\)https://pytorch.org/
parameters of Bi-GCN are updated using stochastic gradient descent, and we optimize the model by Adam algorithm (Kingma and Ba 2014). The dimension of each node’s hidden feature vectors is 64. The dropping rate in DropEdge is 0.2 and the rate of dropout is 0.5. The training process is iterated upon 200 epochs, and early stopping (Yao, Rosasco, and Caponnetto 2007) is applied when the validation loss stops decreasing by 10 epochs. Note that we do not employ SVM-TK on the Weibo dataset due to its exponential complexity on large datasets.

### Overall Performance

Table 2 and Table 3 show the performance of the proposed method and all the compared methods on the Weibo and Twitter datasets, respectively.

First, among the baseline algorithms, we observe that the deep learning methods performs significantly better than those using hand-crafted features. It is not surprising, since the deep learning methods are able to learn high-level representations of rumors to capture valid features. This demonstrates the importance and necessity of studying deep learning for rumor detection.

Second, the proposed method outperforms the PPC_RNN+CNN method in terms of all the performance measures, which indicates the effectiveness of incorporating the dispersion structure for rumor detection. Since RNN and CNN cannot process data with the graph structure, PPC_RNN+CNN ignores important structural features of rumor dispersion. This prevents it from obtaining efficient high-level representations of rumors, resulting in worse performance on rumor detection.

Finally, Bi-GCN is significantly superior to the RvNN method. Since RvNN only uses the hidden feature vector of all the leaf nodes so that it is heavily impacted by the information of the latest posts. However, the latest posts are always lack of information such as comments, and just follow the former posts. Unlike RvNN, the root feature enhancement allows the proposed method to pay more attention to the information of the source posts, which helps improve our models much more.

### Ablation Study

To analyze the effect of each variant of Bi-GCN, we compare the proposed method with TD-GCN, BU-GCN, UD-GCN and their variants without the root feature enhancement. The empirical results are summarized in Figure 3. UD-GCN, TD-GCN, and BU-GCN represent our GCN-based rumor detection models utilize the UnDirected, Top-Down and Bottom-Up structures, respectively. Meanwhile, "root" refers to the GCN-based model with concatenating root features in the networks while "no root" represents the GCN-based model without concatenating root features in the networks. Some conclusions are drawn from Figure 3. First, Bi-GCN, TD-GCN, BU-GCN, and UD-GCN outperforms their variants without the root feature enhancement, respectively. This indicates that the source posts plays an important role in rumor detection. Second, TD-GCN and BU-GCN can not always achieve better results than UD-GCN, but Bi-GCN is always superior to UD-GCN, TD-GCN and BU-GCN. This implies the importance to simultaneously consider both top-down representations from the ancestor nodes, and bottom-up representations from the children nodes. Finally, even when the worst results in Figures 3 are better than those of other baseline methods in Table 2 and 3 by a large gap, which again verifies the effectiveness of graph convolution for rumor detec-
Early Rumor Detection

Early detection aims to detect rumor at the early stage of propagation, which is another important metric to evaluate the quality of the method. To construct an early detection task, we set up a series of detection deadlines and only use the posts released before the deadlines to evaluate the accuracy of the proposed method and baseline methods. Since it is difficult for the PPC RNN+CNN method to process the data of variational lengths, we cannot get the accurate results of PPC RNN+CNN at each deadline in this task, so it is not compared in this experiment.

Figure 4 shows the performances of our Bi-GCN method versus RvNN, SVM-TS, SVM-RBF and DTC at various deadlines for the Weibo and Twitter datasets. From the figure, it can be seen that the proposed Bi-GCN method reaches relatively high accuracy at a very early period after the source post initial broadcast. Besides, the performance of Bi-GCN is remarkably superior to other models at each deadline, which demonstrates that structural features are not only beneficial to long-term rumor detection, but also helpful to the early detection of rumors.

Conclusions

In this paper, we propose a GCN-based model for rumor detection on social media, called Bi-GCN. Its inherent GCN model gives the proposed method the ability of processing graph/tree structures and learning higher-level representations more conducive to rumor detection. In addition, we also improve the effectiveness of the model by concatenating the features of the source post after each GCL of GCN. Meanwhile, we construct several variants of Bi-GCN to model the propagation patterns, i.e., UD-GCN, TD-GCN and BU-GCN. The experimental results on three real-world datasets demonstrate that the GCN-based approaches outperform state-of-the-art baselines in very large margins in terms of both accuracy and efficiency. In particular, the Bi-GCN model achieves the best performance by considering both the causal features of rumor propagation along relationship chains from top to down propagation pattern and the structural features from rumor dispersion within communities through the bottom-up gathering.

Acknowledgments

The authors would like to thank the support of Tencent AI Lab and Tencent Rhino-Bird Elite Training Program. This work is supported by the National Natural Science Foundation of Guangdong Province (2018A030313422), National Natural Science Foundation of China (Grant No. 61773229, No. 61972219) and Overseas Cooperation Research Fund of Graduate School at Shenzhen, Tsinghua University (Grant No. HW2018002).
References


