Modeling microscopic and macroscopic information diffusion for rumor detection

Chen, X.; Zhou, F.; Zhang, F.L.; Bonsangue, M.M.

Received: 7 February 2021 | Revised: 16 March 2021 | Accepted: 11 May 2021

DOI: 10.1002/int.22518

RESEARCH ARTICLE

WILEY

Modeling microscopic and macroscopic information diffusion for rumor detection

Xueqin Chen[1,2] | Fan Zhou[1] | Fengli Zhang[1] | Marcello Bonsangue[2]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China

[2]Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands

Correspondence
Fan Zhou, School of Information and Software Engineering, University of Electronic Science and Technology of China, No. 4, Section 2, North Jianshe Rd, 610054 Chengdu, China.
Email: fan.zhou@uestc.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 62072077; Sichuan Regional Innovation Cooperation Project, Grant/Award Number: 2020YFQ0018; National Key R&D Program of China, Grant/Award Number: 2019YFB1406202

Abstract

Researchers have exerted tremendous effort in designing ways to detect and identify rumors automatically. Traditional approaches focus on feature engineering, which requires extensive manual efforts and are difficult to generalize to different domains. Recently, deep learning solutions have emerged as the de facto methods which detect online rumors in an end-to-end manner. However, they still fail to fully capture the dissemination patterns of rumors. In this study, we propose a novel diffusion-based rumor detection model, called Macroscopic and Microscopic-aware Rumor Detection, to explore the full-scale diffusion patterns of information. It leverages graph neural networks to learn the macroscopic diffusion of rumor propagation and capture microscopic diffusion patterns using bidirectional recurrent neural networks while taking into account the user-time series. Moreover, it leverages knowledge distillation technique to create a more informative student model and further improve the model performance. Experiments conducted on two real-world data sets demonstrate that our method achieves significant accuracy improvements over the state-of-the-art baseline models on rumor detection.

KEYWORDS

knowledge distillation, macroscopic diffusion, microscopic diffusion, rumor detection

1 | INTRODUCTION

The rapid development of Internet technology has spurred various online social platforms (OSN), such as Twitter, Facebook, and so on. These OSNs have provided an environment for free information creation and distribution while substantially changing how people acquire and share information. According to Pew Research in 2017, about 81% of American adults obtain news from online platforms (e.g., news websites/apps, social media, etc.). However, OSNs are a double-edged sword. On the one hand, it brought convenience to our daily life. On the other, the spread of unverified facts such as rumors, fake news, and misinformation have brought negative societal and economic consequences, for example, destabilizing nations, affecting the fairness of competition,[1] and shocking the capital market.[2] Take the more recent event as an example. In the global effort to contain the COVID-19 pandemic,[3] massive misinformation spread on the Internet, and people have been misled to believe that COVID-19 can be cured by ingesting fish tank cleaning products[*] and that 5G networks can generate radiation triggering the virus.[†] Such misinformation not only causes panic among citizens but could potentially undermine collective efforts to control the pandemic. Thus, how to develop a useful rumor detection model has attracted considerable attention from both industry and academic communities.[4–6]

Conventional methods for rumor detection broadly fall into two groups: (1) *hand-crafted feature-based approaches*—mostly identifying and incorporating complicated hand-crafted features for rumor detection, including lexical features,[4,7] syntactic features,[4,8] visual features,[6,9] user profile-related features,[10,11] and social relationship features,[12,13] and so on. Their performance highly depends on the effectiveness of extracted features, which require extensive domain knowledge. (2) *credibility propagation-based approaches*,[9,14,15] which aims to find the truth against conflicting information. These approaches usually leverage the inter-entity relations but heavily rely on the constructed credibility network for high rumor identification accuracy. Recent studies inspired by the successes of deep learning methods in many fields have developed various neural network-based models to learn various feature representations for rumor detection in an end-to-end way. For example, researchers have leveraged recurrent neural networks (RNN) to learn temporal diffusion patterns[5,16] in a sequential learning manner. Nevertheless, such methods fail to capture the complex structural features that are informative signals in identifying rumor spread. The latest approaches have proposed to involve structural information by introducing graph neural networks (GNN)[17–19] to overcome this issue.[20,21] Although these methods have shown performance improvements over the previous methods, they still face several critical limitations. First, most of the existing methods still require a large volume of textual data or a rich collection of users' comments as input.[5,20,21] In addition, previous works focused on either microscopic diffusion patterns that emphasize users' personal retweeting behavior or macroscopic diffusion structures depicting the full rumor in-network diffusion paths.[20,22]

To overcome the limitations mentioned above, in this paper, we propose *M*acroscopic and *M*icroscopic-aware *R*umor *D*etection (MMRD), a novel deep learning-based framework for rumor detection. MMRD models the rumor diffusion from both macroscopic and microscopic perspectives through newly designed encoding components MacroE and MicroE and enhancing the diffusion representations through the cross-learning mechanism. We design a fusion gate to selectively aggregate learned macroscopic and microscopic knowledge and introduce the attention mechanism to merge row-level information to form a unique rumor representation. The rumor prediction is generated based on the learned rumor representation.

Moreover, knowledge distillation technique is applied to further improve the model's detection performance. Our main contributions are summarized into fourfold:

- First, we propose a new model to learn the representation of rumor through modeling the macroscopic and microscopic diffusion. The model is flexible and can be easily integrated into any existing approaches.
- Second, we design two encoding components for macroscopic and microscopic diffusion modeling, respectively, as well as the mechanism to control the information aggregation.
- Third, MMRD employs a powerful technique-knowledge distillation to transfer knowledge from a teacher model to a student model, which further improves the model performance since the student capture more knowledge than the teacher.
- Finally, we conduct extensive evaluations on two benchmark data sets. The experimental results demonstrate that our model significantly outperforms existing baseline methods on rumor detection.

The remainder of the paper is organized as follows. Section 2 reviews the related work of rumor detection. Section 3 formalize the problem of rumor detection and presents some definitions. Section 4 discuss the details of MMRD. The results of the experimental evaluations quantifying the benefits of our approach are presented in Section 5. Finally, the conclusions are discussed in Section 6.

2 | RELATED WORKS

Recently, automatically detecting rumors for OSN becomes requisite due to the increasingly growing fake news and false information and has attracted great attention in both industry and academia. Traditional rumor detection methods mainly focus on extracting hand-crafted features from text contents,[4,7] images,[6,9,23] and social interactions,[7,24,25] which are fed into to discriminative machine learning algorithms to judge whether a piece of information is rumor. These approaches heavily rely on the hand-craft features, but the standard and systematic methodology to design these features are missing. In practice, the conclusions of existing works usually contradict each other because of the discrepancies between different types of datasets. Meanwhile, Gupta et al.[9] introduced a PageRank-like credibility propagation algorithm by encoding users' credibilities and tweets' implications on a user-tweet-event information network. Jin et al.[14] leverage inter-entity implications for credibility propagation and propose a three-layer hierarchical credibility network, which includes news aspects and utilizes a graph optimization framework to infer the event credibility. While comparing with direct classification on hand-crafted features, such credibility propagation-based approaches can exploit the inter-entity relations and achieve robust results—however, their performance strongly affected by the constructed credibility network.

Inspired by the recent success of deep learning in natural language processing (NLP) and computer vision (CV), a few deep learning-based rumor detection methods have emerged and shown significant performance improvement over traditional methods due to their enhanced ability to extract relevant features in an end-to-end. Ma et al.[16] present the first deep learning-based model, which applies an RNN to learn both temporal and linguistic patterns from variable-length time series for rumor detection. Later, they modify the original RNN with a tree-structured RNN to catch the hidden representation from propagation structures and reply text.[5] Liu et al.[22] model user characteristics and propagation paths by combining the advantages of RNN and CNN. Recently, due to the limitation of RNN and CNN in modeling information dissemination, Bian et al.[20] propose a graph convolutional network (GCN)-based mode, which can learn global structural relationships of rumor dispersion. Similarly, Lu et al.[21] use a graph-aware attention network to model the user interaction network, combined with text and user characteristics, for improving the performance of rumor detection. Besides, the nature of multimodal data has inspired researchers to explore effective methods to fuse the content feature and visual feature for rumor detection.[26,27] Moreover, Ma et al.,[28] enlightened by the multitask learning scheme, propose two multitask architectures based on RNNs, which trains the task of stance classification and rumor detection simultaneously. However, these models are still highly dependent on the content (i.e., text and image) features and cannot fully capture the diffusion patterns of rumor detection from both macroscopic and microscopic perspectives.

3 | PRELIMINARIES

In this study, we first borrow the definitions of macro-level and micro-level diffusion prediction from the field of information cascades modeling[29–31] to define macroscopic diffusion and microscopic rumor diffusion, and then give the formalized definition of rumor detection, which are formally defined as follows. In this paper, we use script or italic capital characters (e.g., E, U, $\mathcal{G}$ and $\mathcal{P}$) for sets, bold lowercase characters (e.g., $\mathbf{h}$) for vectors, and bold uppercase characters (e.g., $\mathbf{H}$) for matrices.

Definition 1. *Macroscopic diffusion*. In information cascades modeling, the macro-level diffusion prediction aims at predicting the eventual size of a given cascade. Similarly, the macroscopic diffusion in our work refers to the evolution of the network scale, representing both the change of edges and nodes. We denote the macroscopic diffusion as a diffusion graph $\mathcal{G} = \{U, E\}$, where U is the user set comprising N users, and $E = \{(u_i, u_j)|u_i, u_j \in U\}$ represents a set of edges connecting pairs of users when u_j retweets u_i.

Definition 2. *Microscopic diffusion*. Similar to microlevel diffusion prediction that aims to predict the next infected user, we define the user infected process as microscopic diffusion, that is, who will engage in the information spreading and when this event (retweeting) occurs. In our work, we use a user-time series $\mathcal{P} = \{(u_1, t_1), ..., (u_j, t_j), ..., (u_N, t_N)\}$ to represent the microscopic diffusion, where (u_1, t_1) denotes u_1 created source tweet at time t_1, and the rest of (u_j, t_j) tuples denote user u_j retweet the source tweet at time t_j. Here, all users are in chronological order according to their timestamps.

Definition 3. *User vector*. Each user $u_j \in U$ is represented by a user vector $\mathbf{u}_j \in \mathbb{R}^{F^{user}}$, which is extracted from user profiles, and each dimension of $\mathbf{u}_j$ is related to one kind of profile, such as screen name, description, and so on. (More details of the type of features used in our work can be found in Section 5.1.)

Definition 4. *Rumor detection*. Given a tweet $m_i = \{\mathcal{G}_i, \mathcal{P}_i\}$ within an observation window T, the goal of rumor detection is to learn a classification function $f(m_i)$ to classify m_i as a rumor or nonrumor.

4 | METHODOLOGY

The overall framework of the proposed model MMRD is shown in Figure 1. In particular, it consists of the following main components: (1) the input layer, (2) the macroscopic and microscopic diffusion encoding layer, (3) the fusion gate, (4) the rumor detection layer, and (5) knowledge distillation phase. With this model in mind, we first introduce two essential structural encoding components—*MacroE* and *MicroE*, and then discuss how to generate the unique rumor representation based on the two modules that will preserve both macroscopic and microscopic diffusion properties. Finally, we introduce how to use the knowledge distillation technique to develop a powerful student model for rumor detection.

4.1 | Macroscopic diffusion encoding component

As we depict in Section 3, the macroscopic diffusion of a tweet m_i reflects its diffusion scale. In our work, we cast the macroscopic diffusion modeling as learning the latent structural patterns from the diffusion graph $\mathcal{G}_i$. Inspired by the recent success of GNN in processing the graph structural data, for example, GCN[17,32] and graph attention network (GAT),[18] we implement the macroscopic diffusion encoding component (MacroE) based on vanilla GCN.[17] The vanilla GCN is a multilayer structure that contains several convolution layers, which is defined as:

$$\begin{gathered}\mathbf{H}^{(j+1)} = \sigma\left(\widetilde{\mathbf{A}}\mathbf{H}^{(j)}\mathbf{W}^{(j)}\right)\\ \widetilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I}_N)\mathbf{D}^{-\frac{1}{2}}\end{gathered} \tag{1}$$

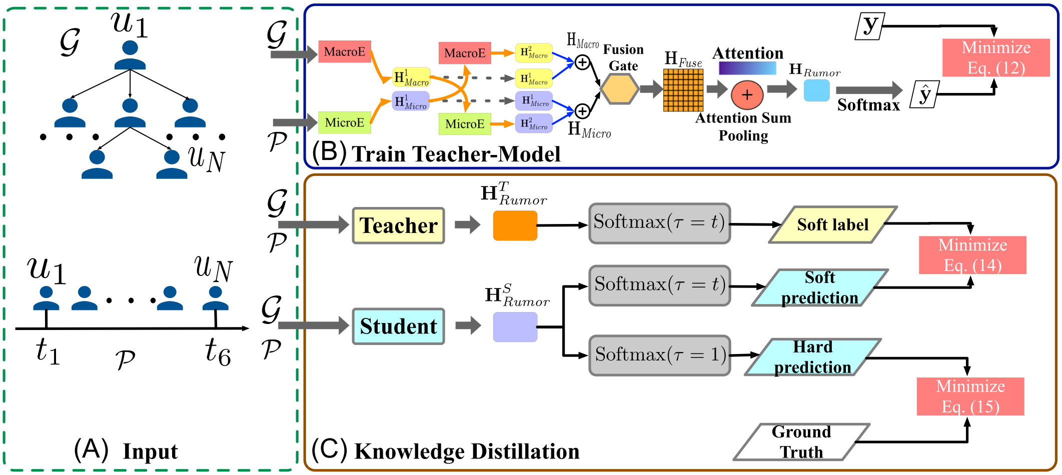


FIGURE 1 Overview of MMRD: (A) inputs of MMRD; (B) normal training process of MMRD; (C) the process of train MMRD with knowledge distillation. MMRD, macroscopic and microscopic-aware rumor detection [Color figure can be viewed at wileyonlinelibrary.com]

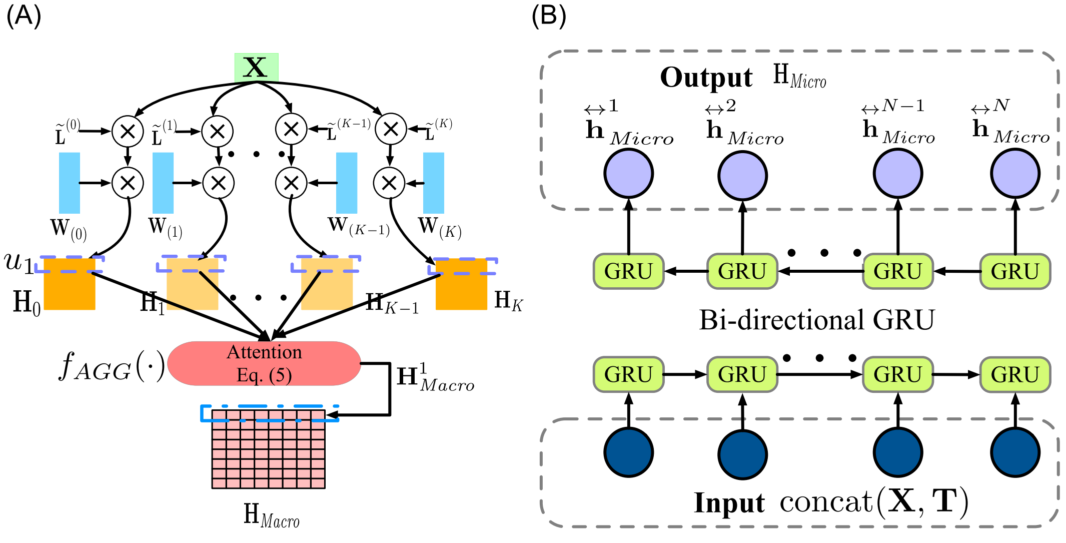
where $\mathbf{H}^{(j)} \in \mathbb{R}^{N \times d_j}$ and $\mathbf{H}^{(j+1)} \in \mathbb{R}^{N \times d_{j+1}}$ are the input and output for layer j, $\mathbf{W}^{(j)} \in \mathbb{R}^{d_j \times d_{j+1}}$ is a trainable weight matrix and $\sigma(\cdot)$ is an activation function (e.g., Relu). $\widetilde{\mathbf{A}}$ is a symmetrically normalized adjacency matrix with self-connections, and $\mathbf{D}$ is a diagonal degree matrix. The adjacency matrix $\mathbf{A}$ and degree matrix $\mathbf{D}$ are expressed as the following:

$$
\begin{aligned}
\mathbf{A}_{ij} &= \begin{cases} 1 & \text{if} \quad (u_i, u_j) \in E \quad \text{and} \quad i \neq j, \\ 0 & \qquad\qquad \text{otherwise}. \end{cases} \\
\mathbf{D}_{ii} &= \sum_j \mathbf{A}_{ij}
\end{aligned}
\tag{2}
$$

The initial input of the first GCN layer $\mathbf{H}^{(0)} = \mathbf{X}$, which is formed by user vectors, i.e., $\mathbf{X} = \{\mathbf{u}_1, \ldots, \mathbf{u}_N\} \in \mathbb{R}^{N \times F^{user}}$. Even the vanilla GCN shows powerful ability in graph embedding, it still faces some limitations: (1) it focuses on undirected graphs rather than the directed graph[33]; and (2) the nodes receive latent representations only from their immediate neighbors, cannot be summarized as higher-order adjacency information.[34,35]

To overcome the aforementioned limitations of GCN in modeling the directed graph and learning higher-order interactions, in this study, we reference the work of CasCN[33] and MixHop[36] and extend the vanilla GCN. Finally, we propose a directed multihop graph convolutional network with attention aggregation as the MacroE (Figure 2A). The convolutional kernel of MacroE is defined as:

$$
\begin{aligned}
\mathbf{H} &= f_{AGG}\left[\sigma(\widetilde{\mathbf{L}}^{(k)}\mathbf{X}\mathbf{W}_{(k)})_{k \in \mathcal{K}}\right] \\
&= \sigma\Big(f_{AGG}\left[\widetilde{\mathbf{L}}^{(0)}\mathbf{X}\mathbf{W}_{(0)} : \widetilde{\mathbf{L}}^{(1)}\mathbf{X}\mathbf{W}_{(1)} \vdots \cdots \vdots\ \widetilde{\mathbf{L}}^{(K)}\mathbf{X}\mathbf{W}_{(K)}\right]\Big) \\
&= \sigma\big(f_{AGG}\,[\mathbf{H}_{(0)} : \mathbf{H}_{(1)} \vdots \cdots \vdots\ \mathbf{H}_{(K)}]\big)
\end{aligned}
\tag{3}
$$



MacroE

MicroE

FIGURE 2 Illustration of diffusion encoding components. (A) MacroE; (B) MicroE [Color figure can be viewed at wileyonlinelibrary.com]

where $\mathbf{X} \in \mathbb{R}^{N\times F}$ is the input feature matrix and $\mathbf{H}_{Macro} = \{\mathbf{h}^1_{Macro}, \ldots, \mathbf{h}^N_{Macro}\} \in \mathbb{R}^{N\times F^{Macro}}$ is the output of MacroE. To capture the directional information from the diffusion graph, we replace the $\widetilde{\mathbf{A}}$ with $\widetilde{\mathbf{L}}$—normalized Laplacian for directed graph. The calculation of $\widetilde{\mathbf{L}}$ is defined as:

$$\begin{aligned}
\mathbf{P} &= (1-\alpha)\frac{\mathbf{E}}{N} + \alpha(\mathbf{D}^{-1}\mathbf{A}),\\
\mathbf{L} &= \Phi^{\frac{1}{2}}(\mathbf{I}-\mathbf{P})\Phi^{-\frac{1}{2}},\\
\widetilde{\mathbf{L}} &= \frac{2}{\lambda_{max}}\mathbf{L} - \mathbf{I}
\end{aligned} \tag{4}$$

where $\mathbf{P}$ is a transition probability matrix, $\mathbf{E} \in \mathbb{R}^{N\times N}$ is an all-one matrix. $\alpha \in (0, 1)$ is an initial probability used to restrict the state transition matrix $\mathbf{D}^{-1}\mathbf{A}$ a strongly connected matrix.[33] Φ is a diagonal matrix with entries $\Phi(v, v) = \phi(v) - \phi(v)$ is the column vector of $\mathbf{P}$,[37] and λ_{max} denotes the largest eigenvalue of $\mathbf{L}$.

$\mathcal{K}$ is a set of integer order powers—the value of $\mathcal{K}$ is from 0 to K, and $\mathbf{W}_k \in \mathbb{R}^{F\times F^{Macro}}$ is the weight matrix for k-hops. $\widetilde{\mathbf{L}}^{(k)}$ denotes the normalized Laplacian matrix $\widetilde{\mathbf{L}}$ multiplied by itself k times, and its value represents the probability connecting path from vertex u_i to vertex u_j in k-hops. Specifically, $\widetilde{\mathbf{L}}^{(0)} = \mathbf{I}$ is an identity matrix. Through the laplacian matrix's multiple powers, MacroE mixes the feature representation of higher-order neighbors in one graph convolutional layer.

$f_{AGG}(\cdot)$ is an aggregation function, which is used to fuse the latent representation from different orders. In most of the existing works,[36,38] $f_{AGG}(\cdot)$ is similar to the pooling methods in CNNs, which can be a mean-pooling function, max-pooling function, or sum-pooling function. However, the distance of the message passing for each node is different, that is, different nodes have different max-orders. In this study, we implement the aggregation function via the order attention mechanism at the node-level. As for each node u_j in the diffusion graph $\mathcal{G}_i$, it has a set of latent representations $\mathbf{h}^j_{(0)}, \ldots, \mathbf{h}^j_{(K)}$ from K-orders. Then the order attention of u_j is calculated as:

$$\begin{aligned}
a^{u_j}_{(k)} &= \frac{\exp\left(\left\langle \mathbf{w}_{u_j}, \tanh\left(\mathbf{W}_{u_j}\mathbf{h}^{u_j}_{(k)} + \mathbf{b}_{u_j}\right)\right\rangle\right)}{\sum_{*=1}^{K}\exp\left(\left\langle \mathbf{w}_{u_j}, \tanh\left(\mathbf{W}_{u_j}\mathbf{h}^{u_j}_{(*)} + \mathbf{b}_{u_j}\right)\right\rangle\right)},\\
\mathbf{h}^{u_j}_{Macro} &= \sum_{k=1}^{K} a^{u_j}_{(k)}\mathbf{h}^{u_j}_{(k)}
\end{aligned} \tag{5}$$

where $\mathbf{W}_{u_j} \in \mathbb{R}^{F^{Macro}\times d}$, $\mathbf{b}_{u_j} \in \mathbb{R}^d$, and $\mathbf{w}_{u_j} \in \mathbb{R}^d$. So that, the aggregation function f_{AGG} is formulated as $f_{AGG} = \{\mathbf{h}^j_{Macro} = \text{Attention}(\mathbf{h}^j_{(0)}, \ldots, \mathbf{h}^j_{(K)}), | j \in \{1, \ldots, N\}\}$. The calculation process of MacroE is outlined in Algorithm 1.

Algorithm 1. Calculation of MacroE (Equation 5).

```
Input: Feature matrix $\mathbf{X}$, normalized laplacian matrix $\widetilde{\mathbf{L}}$, a set of order powers $\mathcal{K}$, and its max-
    value $K = \max(\mathcal{K})$.
Parameters: $\{\mathbf{W}_{(k)}\}_{k \in \mathcal{K}}$.
1: $\mathbf{B} \coloneqq \mathbf{X}$
2: for $k = 0$ to $K$ do
3:    if $k = 0$ then
4:        $\mathbf{B} \coloneqq \mathbf{IB}$
5:    else
6:        $\mathbf{B} \coloneqq \widetilde{\mathbf{L}}\mathbf{B}$
7:    end if
8:    $\mathbf{H}_{(k)} \coloneqq \mathbf{BW}_{(k)}$
9: end for
 /*From step 1 to 9, complete the calculation of $(\widetilde{\mathbf{L}}^{(k)}\mathbf{XW}_{(k)})_{k \in \mathcal{K}}$ in Equation (3)*/
10: $\mathbf{H} \coloneqq f_{AGG}([\mathbf{H}_{(0)}, ..., \mathbf{H}_{(k)}, ..., \mathbf{H}_{(K)}])$ via Equation (5).
11: end return $\sigma(\mathbf{H})$
```

MacroE versus GCN: As depicted above, the convolutional kernel in MacroE for one single order is similar to a single layer of GCN, that is, $\widetilde{\mathbf{L}}^{(k)}\mathbf{XW}_{(k)}$ and $\widetilde{\mathbf{A}}\mathbf{H}^{(j)}\mathbf{W}^{(j)}$, respectively. The main differences between our MacroE and GCN are: (1) we use normalized directed Laplacian $\widetilde{\mathbf{L}}$ to replace the symmetrically normalized adjacency matrix $\widetilde{\mathbf{A}}$ in GCN, which introduces the directional information of edges into the convolution rather than only considering the link information between nodes; and (2) our MacroE can learn high-order information for each node by using one single layer; however, GCN relies on multilayers and may introduce the over-smoothing issue in learning node feature representations.[38]

4.2 | Microscopic diffusion encoding component

The microscopic diffusion encoding component (MicroE) aims to capture temporal patterns from the user engagement time series $\mathcal{P}_i$. Inspired by the success of RNNs in sequential modeling, we employ a bidirectional-GRU (Bi-GRU)[39] as the encoding component, where the hidden states are used to memorize the diffusion history. At each step t_j, Bi-GRU takes the feature vector and previous hidden state as inputs and computes the updated hidden state as:

$$\overleftrightarrow{\mathbf{h}}_j = \mathrm{Bi\text{ - }GRU}(\mathbf{x}_j, \mathbf{h}_{j-1}), \overleftrightarrow{\mathbf{h}}_j \in \mathbb{R}^{F^{Micro}} \tag{6}$$

Then, the output of MicroE module is a sequence of hidden states $\mathbf{H}_{Micro} = \{\overset{\leftrightarrow 1}{\mathbf{h}}_{Micro}, ..., \overset{\leftrightarrow N}{\mathbf{h}}_{Micro}\} \in \mathbb{R}^{N \times F^{Micro}}$.

4.3 | Macroscopic and microscopic cross-learning

After introducing the necessary encoding components, we go to describe how to apply them to learn the latent representations from macroscopic and microscopic diffusion, summarized into two steps. In the first step, we train MacroE and MicroE separately. MacroE takes the feature matrix $\mathbf{X}$, the normalized Laplacian matrix $\widetilde{\mathbf{L}}$ and max-order number K as inputs. As for MicroE, we first represent the infected timestamp of each user into one-hot vector $\mathbf{t}_j \in \mathbb{R}^{d_{time}}$, and then concatenate the timestamp vector matrix $\mathbf{T} = \{\mathbf{t}_1, ..., \mathbf{t}_N\}$ with $\mathbf{X}$ to form the input $\hat{\mathbf{X}}$ for MicroE. Specifically, assume that, the time window is $[0, T]$, and we first split the time window into l disjoint time intervals, and then compute the corresponding time interval for each retweet user u_j as $t_{int}^{j} = \left\lfloor \frac{t_j - t_0}{T/l} \right\rfloor$, where t_0 is the timestamp for the source post user, and t_j is timestamp for u_j. Finally, each user's timestamp is falling into corresponding time intervals and each interval is related to a one-hot embedding, thus, for u_j its timestamp embedding equals to the related time-interval embedding. Note that, in our work, the initial feature matrix $\mathbf{X}$ is extracted from users' profiles. Figure 3 shows a toy example of the model inputs. The outputs of first step are $\mathbf{H}^1_{Macro}$ and $\mathbf{H}^1_{Micro}$, respectively:

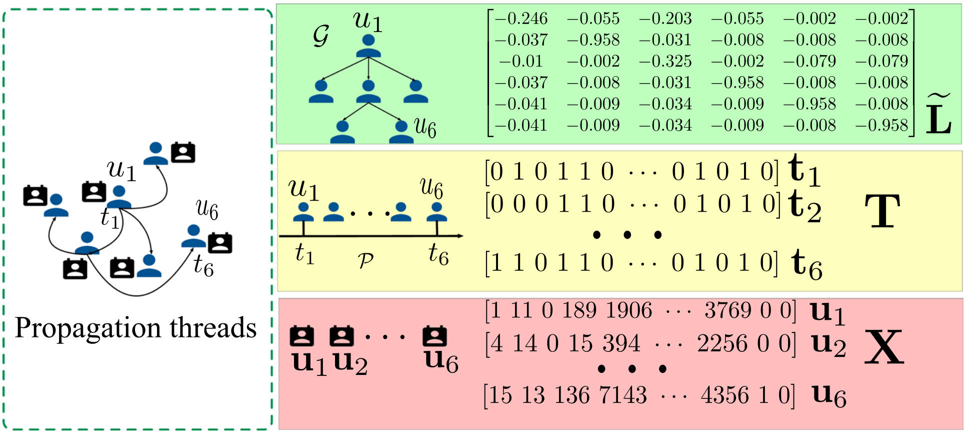
$$\begin{aligned} \mathbf{H}^1_{Macro} &= \text{MacroE}(\mathbf{X}, \widetilde{\mathbf{L}}, K) \\ \mathbf{H}^1_{Micro} &= \text{MicroE}(\text{concat}(\mathbf{X}, \mathbf{T})) \end{aligned} \tag{7}$$

In the second step, we train MacroE and MicroE in a cross-learning manner. Specifically, we use $\mathbf{H}^1_{Macro}$ to train a new MicroE, and vice versa. The outputs of second step are $\mathbf{H}^2_{Macro}$ and $\mathbf{H}^2_{Micro}$:

$$\begin{aligned} \mathbf{H}^2_{Macro} &= \text{MacroE}\left(\mathbf{H}^1_{Micro}, \widetilde{\mathbf{L}}, K\right) \\ \mathbf{H}^2_{Micro} &= \text{MicroE}\left(\mathbf{H}^1_{Macro}\right) \end{aligned} \tag{8}$$

4.4 | Feature fusion via hybrid aggregation layer

We concatenate $\mathbf{H}^1_{Macro}$ with $\mathbf{H}^2_{Macro}$ to form $\mathbf{H}_{Macro} \in \mathbb{R}^{N \times 2F^{Macro}}$, and concatenate $\mathbf{H}^1_{Micro}$ with $\mathbf{H}^2_{Micro}$ to form $\mathbf{H}_{Micro} \in \mathbb{R}^{N \times 2F^{Micro}}$. Thus, for each tweet m_i, we have a macroscopic representation $\mathbf{H}_{Macro}$ and a microscopic representation $\mathbf{H}_{Micro}$. In most of the existing works, after getting $\mathbf{H}_{Macro}$ and $\mathbf{H}_{Micro}$, they will concatenate them directly, however, this operation ignores the different



$$\widetilde{\mathbf{L}} = \begin{bmatrix} -0.246 & -0.055 & -0.203 & -0.055 & -0.002 & -0.002 \\ -0.037 & -0.958 & -0.031 & -0.008 & -0.008 & -0.008 \\ -0.01 & -0.002 & -0.325 & -0.002 & -0.079 & -0.079 \\ -0.037 & -0.008 & -0.031 & -0.958 & -0.008 & -0.008 \\ -0.041 & -0.009 & -0.034 & -0.009 & -0.958 & -0.008 \\ -0.041 & -0.009 & -0.034 & -0.009 & -0.008 & -0.958 \end{bmatrix}$$

$$\mathbf{T}: \quad \begin{aligned} &[0\ 1\ 0\ 1\ 1\ 0\ \cdots\ 0\ 1\ 0\ 1\ 0]\ \mathbf{t}_1 \\ &[0\ 0\ 0\ 1\ 1\ 0\ \cdots\ 0\ 1\ 0\ 1\ 0]\ \mathbf{t}_2 \\ &\quad \bullet\ \bullet\ \bullet \\ &[1\ 1\ 0\ 1\ 1\ 0\ \cdots\ 0\ 1\ 0\ 1\ 0]\ \mathbf{t}_6 \end{aligned}$$

$$\mathbf{X}: \quad \begin{aligned} &[1\ 11\ 0\ 189\ 1906\ \cdots\ 3769\ 0\ 0]\ \mathbf{u}_1 \\ &[4\ 14\ 0\ 15\ 394\ \cdots\ 2256\ 0\ 0]\ \mathbf{u}_2 \\ &\quad \bullet\ \bullet\ \bullet \\ &[15\ 13\ 136\ 7143\ \cdots\ 4356\ 1\ 0]\ \mathbf{u}_6 \end{aligned}$$

FIGURE 3 A toy example of the model inputs [Color figure can be viewed at wileyonlinelibrary.com]

dependence on the two different representations. In our work, to effectively aggregate the learned representations, inspired by the gate mechanism[39] and attention mechanism,[40] we design (1) a fusion gate to fuse $\mathbf{H}_{Macro}$ and $\mathbf{H}_{Micro}$ to form $\mathbf{H}_{Fuse}$, and (2) row-level attention to aggregate features to merge a unique representation $\mathbf{H}_{Rumor}$.

To selectively integrate the important information of two representations, we employ a concise and effective fusion gating mechanism that produces an importance-aware diffusion representation $\mathbf{H}_{Fuse}$ as follows:

$$\begin{aligned} \mathbf{G} &= \text{sigmoid}\left(\mathbf{W}_{gate}^{1}\mathbf{H}_{Macro} + \mathbf{W}_{gate}^{2}\mathbf{H}_{Micro} + \mathbf{b}_{gate}\right) \\ \mathbf{H}_{Fuse} &= \mathbf{G} \odot \mathbf{H}_{Macro} + (1 - \mathbf{G}) \odot \mathbf{H}_{Micro} \end{aligned} \tag{9}$$

where $\mathbf{W}_{gate}^{1}, \mathbf{W}_{gate}^{2} \in \mathbb{R}^{2F' \times 2F'}$, and $\mathbf{b}_{gate} \in \mathbb{R}^{2F'}$. Note that, here $F' = F^{Macro} = F^{Micro}$. $\mathbf{G}$ is used to drop trivial parts of macroscopic representation and add important information from microscopic representation. The rationale behind this design is that the representation fusion $\mathbf{H}_{Fuse} = \{\mathbf{h}_{\text{Fuse}}^{1}, ..., \mathbf{h}_{\text{Fuse}}^{N}\} \in \mathbb{R}^{N \times 2F'}$ would be aware of the different importance of macroscopic and microscopic diffusion.

Then, we merge the row-level information of $\mathbf{H}_{Fuse}$ to form an unique representation $\mathbf{H}_{Rumor}$ for tweet m through attention sum-pooling operation:

$$\begin{aligned} a_j &= \frac{\exp\left(\left\langle \mathbf{w}, \tanh\left(\mathbf{W}_a \mathbf{h}_{Fuse}^{j} + \mathbf{b}_a\right)\right\rangle\right)}{\sum_{*=1}^{N} \exp\left(\left\langle \mathbf{w}, \tanh\left(\mathbf{W}_a \mathbf{h}_{Fuse}^{*} + \mathbf{b}_a\right)\right\rangle\right)}, \\ \mathbf{H}_{Rumor} &= \sum_{j=1}^{N} a_j \mathbf{h}_{Fuse}^{j} \end{aligned} \tag{10}$$

where $\mathbf{W}_a \in \mathbb{R}^{2F' \times d}$, $\mathbf{b}_a \in \mathbb{R}^{d}$ and $\mathbf{w} \in \mathbb{R}^{d}$.

4.5 | Rumor detection and optimization

Subsequently, $\mathbf{H}_{Rumor}$ is used to generate the corresponding binary prediction vector $\hat{\mathbf{y}} = [\hat{y}_0, \hat{y}_1]$, where $\hat{y}_0, \hat{y}_1$ indicate that the prediction probabilities of the label being 0 and 1, respectively, via a fully connected layer and the Softmax function:

$$\hat{\mathbf{y}} = \text{Softmax}(\text{FC}(\mathbf{H}_{Rumor})). \tag{11}$$

In implementation, we train all the model parameters by minimizing the *cross-entropy* between $\hat{\mathbf{y}}$ and $\mathbf{y}$:

$$\mathcal{L} = -\frac{1}{|B|} \sum_{i=1}^{|B|} \sum_{c=0}^{1} y_{i,c} \log \hat{y}_{i,c} + \lambda \ \|\Theta\|_2^2, \tag{12}$$

where $|B|$ is the batch size, $y_{i,c}$ and $\hat{y}_{i,c}$ are the ground truth and predicted results for the ith sample. That is, if the sample belongs to cth class, $\hat{y}_{i,c}$ is 1; otherwise it is 0. $\|\Theta\|_2^2$ is the L_2 regularizer over all the model parameters Θ, and λ is the trade-off coefficient. The optimization can be solved by stochastic gradient descent-based optimization approaches, such as Adam[41] and RAdam.[42] The above computation process of our MMRD model is outlined in Algorithm 2.

Algorithm 2. Training process of MMRD.

```
Input: A set of tweets $\mathcal{M} = \{m_i\}_{i=1}^{|\mathcal{M}|}$, each tweet $m_i = \{\mathcal{G}_i, \mathcal{P}_i\}$, the max-order number $K$.
Output: MMRD-optimized parameters $\Theta$.
initialize $\Theta$
while $\Theta$ has not converged do
    for each tweets batch $B$ do
        for each tweet $M$ in tweets batch $B$ do
            /* (1st) Train MacroE and MicroE separately. */
            1st macroscopic diffusion encoding: $\mathbf{H}^1_{Macro} \leftarrow \mathbf{X}, \widetilde{\mathbf{L}}, K$ via Equation (3);
            1st microscopic diffusion encoding: $\mathbf{H}^1_{Micro} \leftarrow \hat{\mathbf{X}}$ via Equation (6);
            /* (2nd) Cross-learning MacroE and MicroE */
            2nd macroscopic diffusion encoding: $\mathbf{H}^2_{Macro} \leftarrow \mathbf{H}^1_{Micro}, \widetilde{\mathbf{L}}, K$ via Equation (3);
            2nd microscopic diffusion encoding: $\mathbf{H}^2_{Micro} \leftarrow \mathbf{H}^1_{Macro}$ via Equation (6);
            /* Concatenate operation */
            $\mathbf{H}_{Macro} = \text{concat}(\mathbf{H}^1_{Macro}, \mathbf{H}^2_{Macro})$
            $\mathbf{H}_{Micro} = \text{concat}(\mathbf{H}^1_{Micro}, \mathbf{H}^2_{Micro})$
            macroscopic and microscopic representation fusion: $\mathbf{H}_{Fuse} \leftarrow \mathbf{H}_{Macro}, \mathbf{H}_{Micro}$ via Equation (9);
            Attention sum-polling: $\mathbf{H}_{Rumor} \leftarrow \mathbf{H}_{Fuse}$ via Equation (10);
            Estimate the probability $\hat{\mathbf{y}}$ via Equation (11);
        end for
        $\mathcal{L} \leftarrow$ Equation (12);
        $\Theta \leftarrow RAdam(\mathcal{L})$
    end for
end while
```

4.6 | Rumor detection with knowledge distilling

To further improve the model performance on rumor detection task, inspired by knowledge distillation technique[43]—which involves capturing the "dark knowledge" from a teacher model to guide the learning of a student network, has emerged as an essential technique for model improving. We first train a teacher model via Algorithm 2, and then transfer the knowledge from the teacher model to a student model, here in our work, the student model has the same model architecture as the teacher model (self-distillation[44,45]). Before introducing the concrete training procedure of MMRD with knowledge distillation, we first give the definition of the softmax with temperature:

$$q_i = \text{softmax}(\mathbf{H}, \tau) = \frac{exp(\mathbf{H}/\tau)}{\sum_j exp(\mathbf{H}/\tau)} \tag{13}$$

where τ is a temperature that is normally set to 1, using a higher value for temperature τ to produce a softer probability distribution over the class, which brings the advantage that the information carried by the negative label will be relatively amplified, and the model training will pay more attention to the negative label.

The concrete training procedure of the knowledge distillation is listed in Algorithm 4.6, and Figure 1C gives a visualization of Algorithm 4.6. The objective function of the knowledge distillation is a weighted average of two different objective functions. The first loss function is the cross-entropy with the soft targets and it is computed using the same high temperature $\tau = t$ in the softmax of the student model as was used for generating the soft targets from the teacher model.

$$\mathcal{L}_{soft} = -\sum_{i=1}^{|B|} \bar{\mathbf{y}}_i^T \log \bar{\mathbf{y}}_i^S \tag{14}$$

where $\bar{\mathbf{y}}_i^T = \text{softmax}(\text{FC}(\mathbf{H}_{Rumor}^T), \tau = t)$ is soft output from teacher model, and $\bar{\mathbf{y}}_i^S = \text{softmax}(\text{FC}(\mathbf{H}_{Rumor}^S), \tau = t)$ is soft output from student model.

The second loss function is the cross-entropy with the ground truth. This is computed using exactly the same logits in softmax of the student model but at a temperature of 1.

$$\mathcal{L}_{hard} = -\sum_{i=1}^{|B|} \mathbf{y}_i \log \hat{y}_i^S \tag{15}$$

where $\mathbf{y}_i$ is the ground truth and $\hat{y}_i^S = \text{softmax}(\text{FC}(\mathbf{H}_{Rumor}^S), \tau = 1)$ is the hard output of student model. Finally, the objective function of knowledge distillation is:

$$\mathcal{L}_{KD} = (1 - \beta)\mathcal{L}_{soft} + \beta\mathcal{L}_{hard} \tag{16}$$

where β is the balance weight, which always been a considerably lower value since the amplitude of the gradients produced by the scale of the soft output as $1/\tau^2$. This ensures that the relative contributions of the hard and soft targets remain roughly unchanged.[43]

Algorithm 3. Training procedure of MMRD with knowledge distillation.

```
Input: A set of tweets $\mathcal{M} = \{m_i\}_{i=1}^{|\mathcal{M}|}$, each tweet $m_i = \{\mathcal{G}_i, \mathcal{P}_i\}$, the max-order number $K$, temperature $\tau$.
Input: Student-optimized parameters $\Theta$.
Pretrain a Teacher model via Alogrithm 2.
initialize $\Theta$ in Student model.
while $\Theta$ has not converged do
  for each tweets batch $B$ do
    for each tweet $M$ in tweets batch $B$ do
      $\mathbf{H}_{Rumor}^T \leftarrow$ Teacher
      /* Train Student model via Steps 1 to step 9 in Alogrithm 2.*/
      $\mathbf{H}_{Rumor}^S \leftarrow$ Student
      /* Soft outputs from Teacher*/
      $\bar{\mathbf{y}}_i^T = \text{softmax}(\text{FC}(\mathbf{H}_{Rumor}^T), \tau = t)$
      /* Soft outputs from Student*/
      $\bar{\mathbf{y}}_i^S = \text{softmax}(\text{FC}(\mathbf{H}_{Rumor}^S), \tau = t)$
      /* Hard outputs from Student */
      $\hat{y}_i^S = \text{softmax}(\text{FC}(\mathbf{H}_{Rumor}^S), \tau = 1)$
      calculate loss $\mathcal{L}_{KD}$ via Equation (16)
    end for
    $\Theta \leftarrow \text{RAdam}(\mathcal{L}_{KD})$
  end for
end while
```

5 | EXPERIMENTS

We now present the findings from our experimental evaluations. We compare the performance of our MMRD with the state-of-the-art baselines on rumor detection, and we also investigate the effects of different components by comparing several variants of MMRD. Specifically, we would aim at providing empirical evaluations to answer the following research-related questions:

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

RQ2 How does each component of MMRD contribute to the performance?

RQ3 Can MMRD detect rumor at an early stage?

5.1 | Data sets

We conducted our experiments on two real-world data sets: Twitter15 and Twitter16. Both Twitter15 and Twitter16 data sets were collected by Ma et al.[16] Each data set contains a collection of source tweets with its corresponding propagation threads. The original data sets were constructed for multiclass classification, and we removed the tweets labeled as "unverified" or "true rumor" since they were beyond our research interest, and only keep "nonrumor" and "false-rumor" labels as ground truth in both data sets. We built the macroscopic diffusion graph and microscopic diffusion path for each source tweet from its propagation threads. The statistics of the data sets are presented in Table 1. The user profiles were crawled via Twitter API based on the provided user IDs, and for a fair comparison, we follow PPC_RNN + CNN[22] that extracts eight types of characteristics, including, (1) length of a user name; (2) created time of a user account; (3) length of description; (4) followers count; (5) friends count; (6) statuses count; (7) whether the user is verified; and (8) whether the geographical information is enabled.

5.2 | Baselines

We compare our model with a series of state-of-the-art baselines approaches for rumor detection:

- **DTC**[4]: A decision tree-based classification model that combines manually engineered characteristics of tweets to compute the information credibility.
- **SVM-TS**[25]: A linear support vector machine (SVM)-based time series model, which can capture the variation of a broad spectrum of social context information over time by converting the continuous-time stream into fixed time intervals.
- **SVM-RBF**[46]: A SVM-based model that uses radius basis function (RBF) as the kernel, and leverages the handcrafted features of posts for rumor detection.
- **GRU**[16]: An RNN-based model, which learns temporal patterns and content features from user comments for rumor detection.
- **TD-RvNN**[5]: A top-down tree-structured RNN model that explores the importance of propagation structure for rumor detection.

TABLE 1 Statistics of the data sets

Statistic	Twitter15	Twitter16
# source tweets	739	404
# non-rumor	370	199
# rumor	369	205
# users	306,402	168,659
Max. # retweets	2,990	999
Min. # retweets	97	100
Avg. # retweets	493	479
Avg. # time length	743 h	167 h

- **PPC_RNN+CNN**[22]: A model combines RNN and CNN for early rumor detection, which learns the rumor representations through the characteristics of users.
- **Bi-GCN**[20]: A GCN-based model exploring rumor dissemination through bidirectional propagation structures and text contents for rumor detection.
- **GCAN**[21]: A state-of-the-art co-attention network for rumor detection, which learns the rumor representation based on the tweets content and the corresponding retweet users.

5.3 | Parameter settings and evaluation metrics

We implement DTC with Weka,[‡] SVM-TS and SVM-RBF with scikit–learn,[§] and other deep learning-based baselines and our MMRD with Tensorflow.[**] The hyperparameters of baselines are the same as the settings described in the original papers.

Note that, in our work, MMRD only takes the user profiles and timestamps as inputs, and ignores the content features, such as source tweet text and comments, for a fair comparison, we implement some variants for the baselines by changing the initial inputs. Specifically, as for TD-RvNN and Bi-GCN, we use user profile features to replace the comment features, and the variants of these two baselines denoted as TD-RvNN$_{(\text{User})}$ and Bi-GCN$_{(\text{User})}$, respectively. As for GCAN, we remove the source tweet features in the original inputs which termed as GCAN-Text.

The main hyperparameters in our MMRD are tuned as follows. The batch size is 32. The output dimension of MacroE $F^{Macro} = 64$, and the hidden sizes of both the forward GRU and backward GRU units are $F^{Micro} = 32$. The max-order number K is 3. The number of time intervals l is 100 and the embedding size for each timestamp vector d_{time} is 50. The learning rate for both the teacher training phase and knowledge distillation is 0.001 and the balance weight β in distillation is 0.3. The temperature τ in knowledge distillation is 2.5. The training process is iterated upon for 500 epochs but would be stopped earlier if the validation loss does not decrease after 10 epochs. And we randomly choose 70% data for training and the rest of 10% and 20% for validation and testing. In this study, we measured the detection deadline by the number of retweets, that is, the first kth retweets. In the overall performance, the baselines and our MMRD consider the first 40th retweets. We choose accuracy (ACC), precision (Pre), recall (Rec), and F-score (F1) as the evaluation protocols to measure the models' performance in this study.

5.4 | Overall performance (RQ1)

The overall performance is shown in Table 2, from which we can find that our MMRD model consistently outperforms all baselines on both Twitter15 and Twitter16 data sets. In addition to the overall superiority of our model, we have the following observations.

First, compared to the deep learning-based methods, feature-based methods such as DTC, SVM-TS, and SVM-RBF are not competitive because their performance heavily depends on the hand-crafted features. However, designing effective features is time-consuming and requires extensive field-specific knowledge. Furthermore, the performance gain of SVM-TS over DTC lies in its capability of considering time information. On the other hand, SVM-RBF performs slightly better than SVM-TS, suggesting that the kernel-based SVM is better than linear SVM but is still limited to the quality of hand-crafted features.

Second, among all the deep learning-based baselines, GRU, as the early deep learning-based work for rumor detection, performs the worst, primarily because it only relies on temporal-linguistics of the repost sequence but ignores other informative signals such as diffusion structures and user profiles. In addition, both TD-RvNN and Bi-GCN explore the dissemination of rumors on the basis of GRU and learn textual information from replies (i.e., the retweets with comments). However, their performance is not competitive when there are few comments or replies. Bi-GCN generally performs well than TD-RvNN, demonstrating that GCN is a powerful graph learning model compared with tree structure RNN. PPC_RNN+CNN performs relatively well than GRU and TD-RvNN, implying that user-profile information is more informative than text information in rumor detection, the reason is that compared with the replies, in reality, there exist more retweets without any comments, however, the user

TABLE 2 Overall performance comparison of rumor detection on Twitter15 and Twitter16

	Twitter15				Twitter16			
Method	**Acc**	**Pre**	**Rec**	**F1**	**Acc**	**Pre**	**Rec**	**F1**
DTC	0.495	0.494	0.481	0.495	0.561	0.575	0.537	0.562
SVM-TS	0.519	0.519	0.518	0.519	0.693	0.692	0.691	0.692
SVM-RBF	0.535	0.552	0.521	0.536	0.711	0.724	0.709	0.716
GRU	0.580	0.544	0.545	0.544	0.554	0.514	0.516	0.515
TD-RvNN	0.628	0.594	0.616	0.604	0.633	0.619	0.610	0.614
PPC_RNN+CNN	0.691	0.674	0.686	0.679	0.655	0.632	0.651	0.641
Bi-GCN	0.748	0.731	0.759	0.745	0.711	0.709	0.710	0.716
GCAN	0.835	0.825	0.829	0.825	0.823	0.803	0.841	0.822
TD-RvNN$_{(\text{User})}$	0.678	0.671	0.674	0.672	0.661	0.632	0.641	0.636
Bi-GCN$_{(\text{User})}$	0.820	0.846	0.824	0.834	0.814	0.815	0.816	0.816
GCAN-Text	0.683	0.705	0.652	0.678	0.664	0.716	0.579	0.648
MMRD	**0.922**	**0.922**	**0.923**	**0.922**	**0.876**	**0.877**	**0.874**	**0.875**
Improvement	**10.41**%	**11.76**%	**11.34**%	**11.76**%	**6.44**%	**9.22**%	**3.92**%	**6.45**%

Note: The best method is shown in bold, and the second best one is underlined.

information of such retweets is acquirable. The same observations can find when compare Bi-GCN with its variant Bi-GCN$_{(\mathrm{User})}$ and TD-RvNN with TD-RvNN$_{(\mathrm{User})}$.

On the other hand, GCAN takes both text information and user-profile information as input and indeed outperforms other baselines. By comparing GCAN with its variant GCAN-Text, we can find that the performance of GCAN still heavily relies on text information. This is because it models the structural information from the user similarity matrix rather than the retweet network, which may be insufficient in capturing user interactions, and due to the two datasets were existed for a long time, when we try to crawl the user profiles for all users in the datasets, we find that some user accounts do not exist anymore, and it causes difficulties in constructing user similarity graph. Besides that, compare GCAN-Text with Bi-GCN$_{(\mathrm{User})}$, the results of Bi-GCN$_{(\mathrm{User})}$ far exceed GCAN-Text, this observation illustrates the diffusion graph is more powerful than user similarity graph in detecting rumor when ignoring the textual features. To further illustrate that our MMRD indeed significantly outperforms the baselines, we conduct a McNemar's test[47] between our MMRD and the best baseline (GCAN) based on the prediction results on the testing set, and the p-values are 0.001 and 0.013 on Twitter15 and Twitter16, respectively. As $p < 0.05$ on both Twitter15 and Twitter16, we can conclude that MMRD significantly outperforms GCAN.

Our MMRD, in contrast, learns rumor representation from macroscopic and microscopic diffusion without any textual information, suggesting the possibility of detecting rumors by completely exploiting rumors' diffusion patterns. However, the performance of MMRD can be further improved by taking into account other information such as textual information.

5.5 | Ablation study (RQ2)

To answer the RQ2, we conduct several ablation studies from the following perspectives: (1) we first propose five variants of MMRD and compare their performance on both Twitter15 and Twitter16; then, (2) we compare the performance of MMRD in without knowledge distillation and cross distillation settings; finally, (3) we pick up two special parameters to test the model's accuracy change brings by them when changing their value.

5.5.1 | Variants comparison

We conducted an ablation study to explore each component's effect in MMRD by removing a particular component from the original MMRD. Towards that, we derive the following variants of MMRD:

- **-AGG_Atten**: In "-AGG_Atten," we use sum-pooling function to replace the attention aggregation function f_{AGG} in MacroE.
- **-Gate**: In "-Gate," we remove the fusion gate from the MMRD, that is, concatenate $\mathbf{H}_{Macro}$ and $\mathbf{H}_{Micro}$ directly ($\mathbf{H}_{Fuse} = \mathrm{concat}(\mathbf{H}_{Macro}, \mathbf{H}_{Micro})$).
- **-Atten**: In "-Atten," we replace the attention sum-pooling with normal sum-pooling, that is, $\mathbf{H}_{Rumor} = \sum_{j=1}^{N} \mathbf{h}_{Fuse}^{j}$.
- **-GCN**: In "-GCN," we replace the convolution kernel in MacroE with a vanilla GCN layer.
- **-Time**: In "Time," we ignore the timestamp information, that is, the input feature of the first MicroE are user profile features.

TABLE 3 Performance comparison between MMRD and its variants

Method	Twitter15				Twitter16			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
-AGG_Atten	0.854	0.855	0.855	0.855	0.826	0.827	0.824	0.825
-Gate	0.875	0.875	0.874	0.875	0.845	0.845	0.844	0.844
-Atten	0.831	0.835	0.829	0.832	0.795	0.799	0.769	0.784
-GCN	0.851	0.851	0.851	0.851	0.807	0.807	0.806	0.807
-Time	0.878	0.878	0.879	0.878	0.845	0.863	0.839	0.851
MMRD	**0.922**	**0.922**	**0.923**	**0.922**	**0.876**	**0.877**	**0.874**	**0.875**

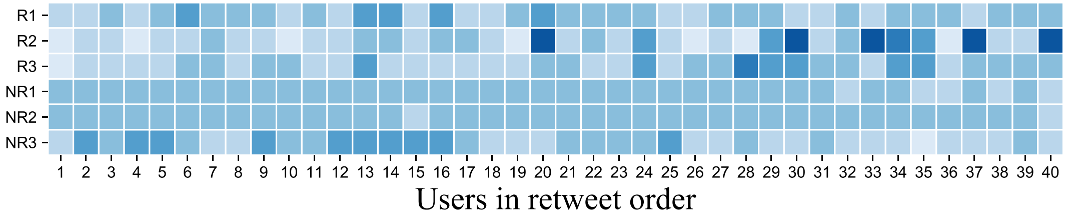


FIGURE 4 Visualization of attention weights in attention sum-pooling, which randomly choose three rumors and three nonrumors from Twitter15. Dark colors refer to a higher value [Color figure can be viewed at wileyonlinelibrary.com]

The results of the ablation study are summarized in Table 3, where we can observe that:

(1) The accuracy of "-Atten" remarkably decreases compared with other variants, which indicates that user-level attention sum-pooling can learn the importance of each user in rumor diffusion since it allocates different significance to each row (that correlated to a specific user) of $\mathbf{H}_{Fuse}$. The visualization of the attention weights is depicted in Figure 4, which further proves the effectiveness. From Figure 4, we also find that the later users are more critical in rumor spreading, which confirms the hypothesis that rumors can spread deeper than nonrumors.[48] (2) Using the fusion gate to control the dependency on macroscopic diffusion and microscopic diffusion will improve the model performance as achieved by the "-Gate." (3) The results of "-GCN" demonstrate that multihop and directional information are essential for macroscopic diffusion modeling, and the performance of "-AGG_Atten" worse than MMRD, which further demonstrates that as for each node, their order-dependency is different. (4) As for "-Time," it shows the importance of the timestamp information in capturing microscopic diffusion.

5.5.2 | Performance on knowledge distillation

In our work, one of the most important components is the use of knowledge distillation to enhance model performance. To test the performance of knowledge distillation (for briefly, simplify as KD), in this section, we conduct experiments on removing KD and using cross KD, respectively.

Figure 5 shows the results when removing the KD, we find that, after removing KD, although the model still can achieve better performance compared with the baselines in Table 2, it can be further improved by using KD to transfer knowledge from a teacher model to

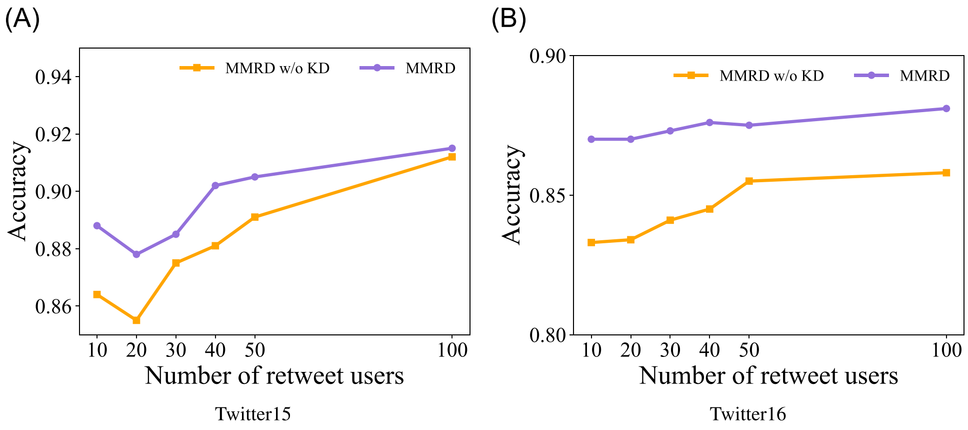


FIGURE 5 The effectiveness of knowledge distillation. The number of observed retweet users per source tweet varies from 10 to 100, and we plot the corresponding detection accuracy of MMRD with and without knowledge distillation. (A) Twitter15; (B) Twitter16. MMRD w/o KD, denotes MMRD without knowledge distillation [Color figure can be viewed at wileyonlinelibrary.com]

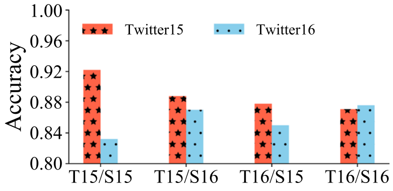


FIGURE 6 Cross knowledge distillation. The number of observed retweet users per source tweet sets to 40. Each bar represents the detection accuracy and the labels of the *x*-axis denote the data sets used when training the teacher model and student model. For example, "T15" and "T16" denote that we train the teacher model on Twitter15, and Twitter16, respectively; "S15" and "S16" means that we learn the student model via distilling knowledge of Twitter15 and Twitter16, respectively [Color figure can be viewed at wileyonlinelibrary.com]

a student model. Besides that, the effect of KD is more remarkable on the Twitter16 data set, it yields a large performance interval between "MMRD" and "MMRD w/o KD."

Figure 6 shows the comparison between different strategies of cross KD. Specifically, we train the teacher model and student model based on different datasets and then test the student's performance on both Twitter15 and Twitter16 datasets. For example, "T15/S16" means we first train a Teacher model "T15" on Twitter15 data set and then distill the model on Twitter16 to get a student model "S16", and finally use the "S16" model to perform rumor detection on Twitter15 and Twitter16, respectively. From Figure 6, we observe that the performance of MMRD is much better when both the Teacher model and Student model train on the same data set, this is because of some data set-specific reasons, such as diffusion scale, the number of non-exist users, user-specific feature (e.g., create time), and so on.

5.5.3 | Parameter analysis

From all parameters in MMRD, we choose two special parameters to conduct parameter analysis experiments—the value of max-order K and the time embedding size d_{time}. The results

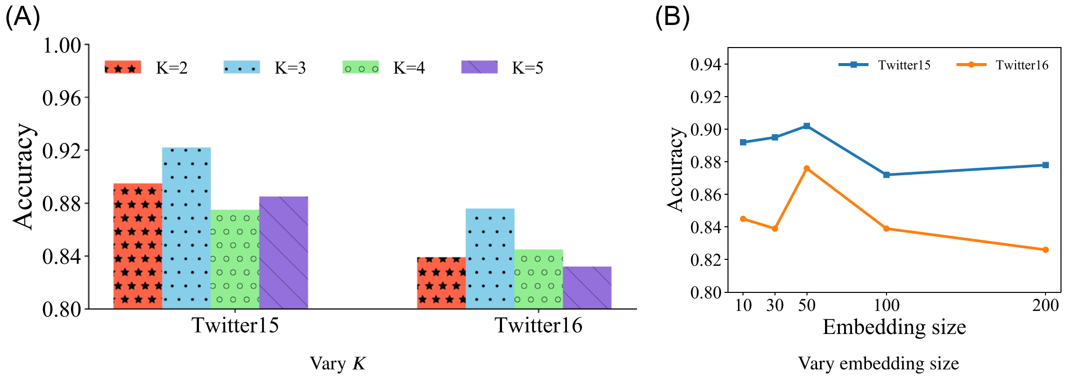


FIGURE 7 Results of parameter analysis on Twitter15 and Twitter16 when the number of observed retweet users per source tweet sets to 40. (A) Performance on different max-order value K, ranging from 2 to 5. (B) Performance on different embedding size d_{time} of timestamp vector $\mathbf{T}$ (A) Vary K; (B) Vary embedding size [Color figure can be viewed at wileyonlinelibrary.com]

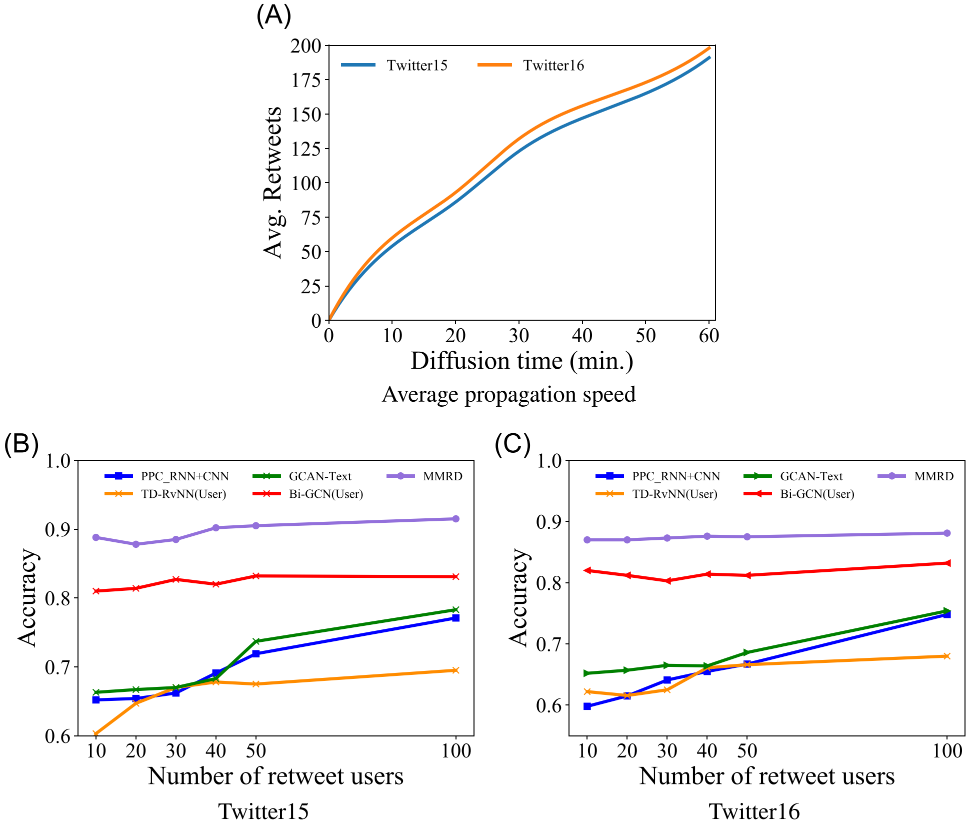


FIGURE 8 Evaluations on early rumor detection. (A) The average propagation speed of tweets calculated based on Twitter15 and Twitter16 datasets. (B) and (C) plot the detection accuracy when the number of observed retweet users per source tweet are in the range of [10, 20, 30, 40, 50, 100] on Twitter15 and Twitter16, respectively (A) Average propagation speed; (B) Twitter15; and (C) Twitter16 [Color figure can be viewed at wileyonlinelibrary.com]

shows in Figure 7. From both Figure 7A,B, we find that by blindly increase the number of K and d_{time}, the model accuracy not improve, instead, decreased. And when set $K = 3$ and $d_{time} = 50$, the model achieves the best performance. Moreover, the embedding size d_{time} with small values achieve better performance than large values. And Figure 7A also demonstrates that take the higher-order of node interaction into consideration is useful when modeling macroscopic diffusion of tweets.

5.6 | Early detection (RQ3)

Detecting rumors as early as possible is crucial for public opinion control. Figure 8A shows the average propagation speed of messages on twitter calculated based on Twitter15 and Twitter16. We find that within 60 min, both Twitter15 and Twitterr15 have a diffusion speed near 190 retweets. And when the time is extremely small, that is, within 30 min, the average retweets of both two datasets are close to 100. So, to investigate the performance of models on identifying rumors at an early stage, here, we consider the number of observed retweet users per source tweet from the list [10, 20, 30, 40, 50, 100]. Besides, for a fair comparision, we choose user profile-based mtheods, that is, "TD-RvNN$_{\text{(User)}}$," "PPC_RNN+CNN," "Bi-GCN$_{\text{(User)}}$," and "GCAN-Text" as contrast methods. Figure 8B,C show the performance comparison of early-stage detection between our MMRD and selected baselines. We can see that MMRD performs better, especially when there are only a few observations, that is, MMRD achieves almost 89% and 87% accuracy on Twitter15 and Twitter16, respectively, even with only 10 retweet user observations.

6 | CONCLUSION

This paper proposed a novel rumor detection model named MMRD, which can effectively and efficiently summarize a unique representation for each rumor propagation through capturing the dissemination patterns from both macroscopic and microscopic diffusion levels. Simultaneously, MMRD leverages the knowledge distillation technique to transfer knowledge from a pretraining teacher model to a student model which further improves the model detection performance. The experimental results based on two real-world Twitter data sets demonstrate that our method achieves state-of-the-art performance on rumor detection and also effective in detecting rumors at an early stage. Besides that, MMRD detects rumor via learning its spreading process, which can help us to develop rumor spreading models.[49] In future work, we plan to extend the summarized representation from the proposed model to other downstream applications, such as link prediction,[50] micro- and macro- information cascades prediction,[33] and so on. Also, we plan to modify the MMRD framework for the spatial-temporal modeling task.[51] Furthermore, incorporating more side knowledge and attributes into the proposed model, including text content and images, and so on, to further improve the detection accuracy is worthy of investigating.

ACKNOWLEDGMENTS

This study was supported by National Natural Science Foundation of China (Grant no. 62072077), Sichuan Regional Innovation Cooperation Project (Grant no. 2020YFQ0018), and National Key R&D Program of China (2019YFB1406202).

ENDNOTES

[*] https://www.bbc.com/news/52012242

[†] https://www.bbc.com/news/52168096

[‡] https://www.cs.waikato.ac.nz/ml/weka/

[§] https://scikit-learn.org/

[¶] https://www.tensorflow.org/

ORCID

Xueqin Chen https://orcid.org/0000-0003-1538-3713
Fan Zhou https://orcid.org/0000-0002-8038-8150
Fengli Zhang https://orcid.org/0000-0003-2300-8817
Marcello Bonsangue https://orcid.org/0000-0003-3746-3618

REFERENCES

1. Allcott H, Gentzkow M. Social media and fake news in the 2016 election. *J Econ Perspectives*. 2017;31(2): 211-236.
2. DiFonzo N, Bordia P. Rumor and prediction: making sense (but losing dollars) in the stock market. *Organ Behav Hum Perform*. 1997;71(3):329-353.
3. Ashraf S, Abdullah S. Emergency decision support modeling for COVID-19 based on spherical fuzzy information. *Int J Intell Syst*. 2020;35(11):1601-1645.
4. Castillo C, Mendoza M, Poblete B. Information Credibility on Twitter. In: *WWW'*11. *Proceedings of the* 20*th International Conference on World Wide Web*. ACM; 2011:675-684.
5. Ma J, Gao W, Wong KF. Rumor detection on twitter with tree-structured recursive neural networks. In: *ACL'*18. *Proceedings of the* 56*th Annual Meeting of the Association for Computational Linguistics*. ACL; 2018: 1980-1989.
6. Zhang Z, Zhang Z, Li H. Predictors of the authenticity of Internet health rumours. *Health Information & Libraries*. 2015;32(3):195-205.
7. Kwon S, Cha M, Jung K, Chen w, Wang Y. Prominent features of rumor propagation in online social media. In: *ICDM'*13. 2013 *IEEE* 13*th International Conference on Data Mining*. IEEE; 2013:1103-1108.
8. Hassan A, Qazvinian V, Radev D. What's with the Attitude? Identifying Sentences with Attitude in Online Discussions. In: *EMNLP'*10. *Proceedings of the* 2010 *Conference on Empirical Methods in Natural Language Processing*. ACL; 2010:1245-1255.
9. Gupta M, Zhao P, Han J. Evaluating event credibility on twitter. In: *SIAM'*12. *Proceedings of the* 2012 *SIAM International Conference on Data Mining*. SIAM; 2012:153-164.
10. Shu K, Wang S, Liu H. Understanding user profiles on social media for fake news detection. In: *MIPR'*18. 2018 *IEEE Conference on Multimedia Information Processing and Retrieval*. IEEE; 2018:430-435.
11. Shu K, Zhou X, Wang S, Zafarani R, Liu H. The role of user profiles for fake news detection. In: *ASONAM'*19. *Proceedings of the* 2019 *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM; 2019:436-439.
12. Shu K, Wang S, Liu H. Beyond news contents: the role of social context for fake news detection. In: *WSDM'*19. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM; 2019:312-320.
13. Yang Y, Niu K, He Z. Exploiting the topology property of social network for rumor detection. In: *JCSSE'*15. 12*th International Joint Conference on Computer Science and Software Engineering*. IEEE; 2015:41-46.
14. Jin Z, Cao J, Jiang YG, Zhang Y. News credibility evaluation on microblog with a hierarchical propagation model. In: *ICDM'*14. 2014 *IEEE International Conference on Data Mining*. IEEE; 2014:230-239.
15. Jin Z, Cao J, Zhang Y, Luo J. News verification by exploiting conflicting social viewpoints in microblogs. In: *AAAI'*16. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press; 2016: 2972-2978.

16. Ma J, Gao W, Mitra P, et al. Detecting rumors from microblogs with recurrent neural networks. In: *IJCAI'*16. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press; 2016:3818-3824.
17. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: *ICLR'*17. 5*th International Conference on Learning Representations*. ICLR; 2017.
18. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph Attention Networks. In: *ICLR'*18. 6*th International Conference on Learning Representations*. ICLR; 2018.
19. Zhou F, Yang Q, Zhong T, Chen D, Zhang N. Variational graph neural networks for road traffic prediction in intelligent transportation systems. *IEEE Trans Industr Inform*. 2021;17(4):2802-2812.
20. Bian T, Xiao X, Xu T, et al. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. In: *AAAI'*20. *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press; 2020:549-556.
21. Lu YJ, Li CT. GCAN: graph-aware co-attention networks for explainable fake news detection on social media. In: *ACL'*20. *Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics*. ACL; 2020:505-514.
22. Liu Y, Wu YFB. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: *AAAI'*18. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI; 2018:354-361.
23. Zhou F, Cao C, Zhong T, Geng J. Learning meta-knowledge for few-shot image emotion recognition. *Expert Syst Appl*. 2021;168:114274.
24. Liu X, Nourbakhsh A, Li Q, Fang R, Shah S. Real-time rumor debunking on twitter. In: *CIKM'*15. *Proceedings of the* 24*th ACM International on Conference on Information and Knowledge Management*. ACM; 2015:1867-1870.
25. Ma J, Gao W, Wei Z, Lu Y, Wong K. Detect rumors using time series of social context information on microblogging websites. In: *CIKM'*15. *Proceedings of the* 24*th ACM International on Conference on Information and Knowledge Management*. ACM; 2015:1751-1754.
26. Jin Z, Cao J, Guo H, Zhang Y, Luo J. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In: *MM'*17. *Proceedings of the* 25*th ACM International Conference on Multimedia*. ACM; 2017:795-816.
27. Khattar D, Goud JS, Gupta M, Varma V. Mvae: multimodal variational autoencoder for fake news detection. In: *WWW'*19. *The World Wide Web Conference*. ACM; 2019:2915-2921.
28. Ma J, Gao W, Wong K. Detect rumor and stance jointly by neural multi-task learning. In: *WWW'*18. *Companion Proceedings of the The Web Conference* 2018. *IW3C2*; 2018:585-583.
29. Chen X, Zhang K, Zhou F, Trajcevski G, Zhong T, Zhang F. Information Cascades Modeling via Deep Multi-Task Learning. In: *SIGIR'*19. *Proceedings of the* 42*nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM; 2019:885-888.
30. Zhou F, Xu X, Trajcevski G, Zhang K. A survey of information cascade analysis: models, predictions, and recent advances. *ACM Comput Surv*. 2021; 54(1).
31. Zhou F, Xu X, Zhang K, Trajcevski G, Zhong T. Variational information diffusion for probabilistic cascades prediction. In: *IEEE*; 2020:1618-1627.
32. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: *NeurIPS'*16. *Proceedings of the* 30*th International Conference on Neural Information Processing Systems*. Curran Associates Inc.; 2016:3844-3852.
33. Chen X, Zhou F, Zhang K, Trajcevski G, Zhong T, Zhang F. Information diffusion prediction via recurrent cascades convolution. In: *ICDE'*19. *IEEE* 35*th International Conference on Data Engineering*. IEEE; 2019: 770-781.
34. Lee JB, Rossi RA, Kong X, Kim S, Koh E, Rao A. Higher-order graph convolutional networks. arXiv preprint. arXiv:1809.07697; 2018.
35. Ma Y, Wang S, Aggarwal CC, Tang J. Graph convolutional networks with eigenpooling. In: *KDD'*19. *Proceedings of the* 25*th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM; 2019:723-731.

36. Abu-El-Haija S, Perozzi B, Kapoor A, et al. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In: *PMLR'*19. *Proceedings of the* 36*th International Conference on Machine Learning. PMLR*; 2019:21-29.
37. Chung F. Laplacians and the Cheeger inequality for directed graphs. *Annals Combin*. 2005;9(1):1-19.
38. Lei F, Liu X, Jiang J, Liao L, Cai J, Zhao H. Graph convolutional networks with higher-order pooling for semisupervised node classification. *Concurr Computat Pract Exp*. 2020:e5695
39. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: *NIPS'*14. *NIPS* 2014 *Workshop on Deep Learning*. Curran Associates Inc.; 2014.
40. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: *NeurIPS'*17. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*. Curran Associates Inc.; 2017:6000-6012.
41. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: *ICLR'*15. 3*rd International Conference on Learning Representations*. ICLR; 2015.
42. Liu L, Jiang H, He P, et al. On the variance of the adaptive learning rate and beyond. In: *ICLR'*20. *Proceedings of the Eighth International Conference on Learning Representations*. ICLR; 2020.
43. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. In: *NIPS'*15. *NIPS Deep Learning and Representation Learning Workshop*; 2015.
44. Hou Y, Ma Z, Liu C, Loy CC. Learning lightweight lane detection CNNs by self attention distillation. In: *ICCV'*19. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. ICCV; 2019:1013-1021.
45. Zhang L, Song J, Gao A, Chen J, Bao C, Ma K. Be your own teacher: improve the performance of convolutional neural networks via self distillation. In: *ICCV'*19. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. ICCV; 2019:3713-3722.
46. Yang F, Liu Y, Yu X, Yang M. Automatic detection of rumor on Sina Weibo. In: *MDS'*12. *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM; 2012.
47. Dietterich TG. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computat*. 1998;10(7):1895-1923.
48. Vosoughi S, Roy D, Aral S. The spread of true and false news online. *Science*. 2018; 359(6380): 1146-1151.
49. Ai S, Hong S, Zheng X, Wang Y, Liu X. CSRT rumor spreading model based on complex network. *Int J Intell Syst*. 2021;36(5):1903-1913.
50. Bastani S, Jafarabad AK, Zarandi MHF. Fuzzy models for link prediction in social networks. *Int J Intell Syst*. 2013;28(8):768-786.
51. Xiao Y, Yin H, Zhang Y, Qi H, Zhang Y, Liu Z. A dual-stage attention-based Conv-LSTM network for spatio-temporal correlation and multivariate time series prediction. *Int J Intell Syst*. 2021.

How to cite this article: Chen X, Zhou F, Zhang F, Bonsangue M. Modeling microscopic and macroscopic information diffusion for rumor detection. *Int J Intell Syst*. 2021;36:5449-5471. https://doi.org/10.1002/int.22518