RESEARCH ARTICLE

Rumor Detection with Self-supervised Learning on Texts and Social Graph

Yuan GAO[1], Xiang WANG(✉)[1], Xiangnan HE(✉)[1], Huamin FENG[2], Yongdong ZHANG[1]

1   School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China

2   Beijing Electronic Science and Technology Institute, Beijing 102627, China

Abstract   Rumor detection has become an emerging and active research field in recent years. At the core is to model the rumor characteristics inherent in rich information, such as propagation patterns in social network and semantic patterns in post content, and differentiate them from the truth. However, existing works on rumor detection fall short in modeling heterogeneous information, either using one single information source only (*e.g.,* social network, or post content) or ignoring the relations among multiple sources (*e.g.,* fusing social and content features via simple concatenation).

Therefore, they possibly have drawbacks in comprehensively understanding the rumors, and detecting them accurately. In this work, we explore contrastive self-supervised learning on heterogeneous information sources, so as to reveal their relations and characterize rumors better. Technically, we supplement the main supervised task of detection with an auxiliary self-supervised task, which enriches post representations via post self-discrimination. Specifically, given two heterogeneous views of a post (*i.e.,* representations encoding social patterns and semantic patterns), the discrimination is done by maximizing the mutual information between different views of the same post compared to that of other posts. We devise cluster-wise and instance-wise approaches to generate the views and conduct the discrimination, considering different relations of information sources. We term this framework as *Self-supervised Rumor Detection* (SRD). Extensive experiments on three real-world datasets validate the effectiveness of SRD for automatic rumor detection on social media.

Keywords   Rumor Detection, Graph Neural Networks, Self-supervised Learning, Social Media

1 Introduction

Social media platforms (*e.g.,* Twitter, Facebook, Sina Weibo) have emerged as a major source of information, where users can easily view, post, forward, and comment on any content at little price. However, it also causes the proliferation of rumors, which encode misinformation and disinformation. The massive spread of rumor circulation has become a serious threat, influencing the societal or political opinions of the public. Taking the 2016 US presidential election as an example, the inflammatory news that favored two nominees was shared over 37 million times on Facebook [1], while around 19 million malicious bot accounts tweeted or retweeted supporting two nominees on Twitter [2]. Therefore, it is of great importance to detect rumors accurately and debunk rumors early before they reach a broad audience.

Identifying and modeling the patterns of rumor posts has become the theme, so as to differentiate the rumors from the truths. Towards this end, multiple information sources are leveraged to profile the rumors and truths. Among them, two informative and useful sources are (1) social network, which consists of user-user social connections and user-post interaction connections, reflecting how information flows and propagates along with these connections; and (2) post content, which are typically composed of text pieces and often associated with pictures or videos, that encode rich semantics. Here we focus mainly on these semantic and propagation patterns.

Existing works on rumor detection follow a supervised learning scheme — generating post representations from these information sources and feeding them into a supervised learning model guided by the ground-truth labels. Learning post representations

roughly falls into two types: (1) using one information source individually; (2) fusing two information sources. Nonetheless, both types of methods suffer from some inherent limitations:

- There are numerous efforts on modeling either post content or social network solely, but failing to model them simultaneously. Specifically, a line considers the semantic patterns (*e.g.,* lexical, syntactic, sentiment, topic-level features) in post content [3, 4], while another line focuses on the propagation patterns in social network [5, 6], in order to generate high-quality representations of rumors. For example, earlier work [7] proposes to formulate the rumor propagation as a sequence via a recursive neural network. More recently, Bi-GCN [5] employs graph neural network (GNN) over the holistic social network to perform the information propagation and aggregation among relevant posts. While being effective from the perspective of modeling individual information source, these methods leave other supplementary information untouched, which degrade model effectiveness.

- Some works [8, 9] explore how to integrate multiple information sources to characterize rumors more comprehensively. For example, CGAT [8] combines the representations encoding propagation and semantic patterns together via concatenation and pooling operators. Nevertheless, such integration is of limited ability to model the relations between these co-occurring heterogeneous patterns. It has been proved that information with different contents has different audiences and different propagation paths which makes different temporal patterns and diffusion speeds [10]. Taking the 2016 US presidential elec-
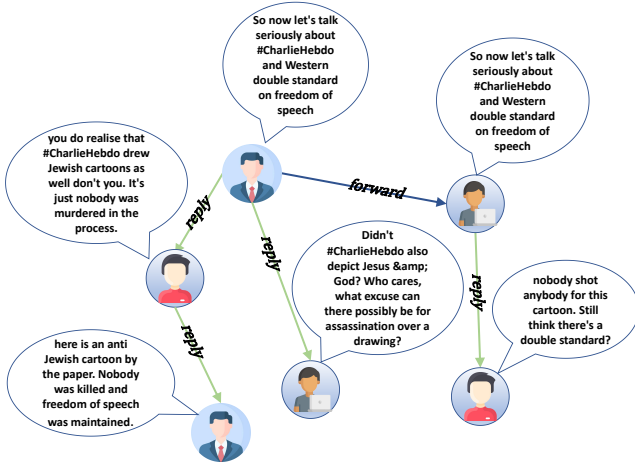
Fig. 1 Real rumor in PHEME dataset [11], which is expressed as propagation tree with text content.

tion as an example, given the content of a post supporting one nominee, we could possibly infer the audiences of interest and their sentiments, and further imagine its propagation among these audiences; analogously, we could estimate the coarse-grained topic of a post, based on its information flow. Hence, ignoring the relations falls short in revealing hidden patterns and results in suboptimal representations of rumors.

To overcome the limitations of prior studies, we explore self-supervised learning (SSL) [12–14] on heterogeneous information sources, to reveal their relations and characterize rumors better. Technically, we supplement the main supervised learning task of rumor detection with an auxiliary self-supervised learning task of post self-discrimination. These two tasks are founded upon the same post representations — we apply the GNN encoder over the social network and aggregate the information from the neighbors, so as to inject the propagation patterns into the social representation of a post; meanwhile, we employ the CNN encoder to generate the semantic representations from a post's content. The SSL task is composed of two modules: (1) Data

augmentation: we devise instance-wise and cluster-wise augmentation operators, and leverage them to derive different views of a post from its social and semantic representations; (2) Contrastive learning: we encourage the agreement between different views of the same post to be maximized, while minimizing the agreement between different posts. Such an SSL task allows us to explicitly measure the relations among different information sources, and guides the learning of post representations. We term this framework as *Self-Supervised Rumor Detection* (SRD). Extensive experiments are conducted on three real-world datasets. Experimental results from Twitter and Weibo demonstrate superior performance over the state-of-the-art methods like Bi-GCN [5]. Our contribution can be summarized as follows:

- We integrate the propagation and semantic patterns of posts via self-supervised learning. To the best of our knowledge, we are the first to leverage self-supervised learning for rumor detection on social media.

- We propose cluster-wise and instance-wise discrimination as the self-supervised learning task to fulfill the potential of heterogeneous data.

- Extensive experiments on three real-world datasets demonstrate the superiority of our proposed framework. We will release our codes upon acceptance.

2 Related Work

In this section, we review the work on rumor detection, graph convolutional network, and self-supervised learning.

2.1 Rumor Detection

Recently, automatic rumor detection comes into public attention. Traditional methods mainly focus on machine learning algorithms such as decision trees and SVM, which utilize manually extracted discrete features to determine if the sample given is worth trusting. Castillo et al. [15] identifies four kinds of features from Twitter, message, user, topic, and propagation, respectively. Following-up works mainly focus on characterizing one or a mixture of the 4 types of features. Yang et al. [16] studies two new features, the client program used and event location, and extends the original work to Chinese. Ma et al. [17] takes time stamps for news events into account and designs a dynamic series-time structure to model the changes or the trends of individual messages along the life cycle. Liu et al. [18] points out the importance of real-time prediction, and found some features are more important for early detection while others begin dominating in later stages. Ma et al. [19] adopts propagation trees to represent the propagation of each source tweet, and captures sub-structures that are indicative of rumors by estimating the similarity between propagation trees via propagation tree kernel learning. The performance of these models largely relies on the quality of feature engineering.

With the rise of deep learning and neural network, the focus has shifted from feature engineering to model learning. Ma et al. [20] employs Recurrent Neural Network (RNN) to model news event as a sequence to better capture contextual information. Jin et al. [21] fuses features from textual, visual, and social context contents, and utilizes attention mechanism to align features in RNN. Chen et al. [22] embeds attention mechanism to RNN to enable importance focus that varies over time. Some researchers explore adding additional features to the model. Guo et al. [23] adopt some social features and design a hierarchical social attention network. And Li et al. [24] add user profiles to evaluate the user credibility. Ma et al. [7] proposes tree-structured Recursive Neural Networks (RvNN) to make predictions in a hierarchical structure with features extracted from both text contents and propagation structures. Yu et al. [25] adopts a Convolutional Neural Networks (CNN) based model to automatically obtain key features from the representation learned from an unsupervised method - paragraph vector. Kumar et al. [26] introduces both a CNN and a bidirectional LSTM ensembled network with an attention mechanism to do rumor detection. Similarly, Liu et al [27] detects fake news using a combination of recurrent and convolutional networks. Rao et al [28] pre-trains a two-layer BERT, which takes comments as auxiliary features for fine-tuning. Recently, adversarial learning is proved to be helpful with the robustness of the model. Song et al [29] devises adversarial learning to improve the vulnerability of detection models. DropAttack [30] is a new adversarial training method, which allows a certain weight parameter of the model to be attacked with a certain probability. These methods make much progress in misinformation identification, but they neglect the global structural features of rumor propagation.

2.2 Graph Convolutional Network

Inspired by the great success of Convolutional Neural Network (CNN), Graph Neural Networks (GNNs) begin to emerge in supervised or semi-supervised tasks like node classification [31], link prediction [32] and graph classification [33], as well as unsupervised tasks like network embedding [34] and graph generation [35].

Graph Neural Network can be categorized into two directions: the spectral-based approaches and the spatial-based approaches [36]. The spectral-based approaches are based on spectral graph theory, they inherit the idea from graph signal processing and propose variant graph convolution kernels. The spectral-based methods are powerful but most of them are in transductive settings. The spatial-based methods construct convolutional filters by looking into the spatial relationship between nodes. Since their focus is on training a function that aggregates neighborhood information to get the representation of the current node, spatial-based methods generalize GNN to inductive learning.

Kipf et al. [31] proposes a fast approximate convolution on graphs, which becomes one of the most representative works in Spectral GCN. Niepert et al. [37] discusses the equivalence of neighborhood graph and receptive field. Hamilton et al. [38] proposes GraphSAGE, an inductive framework that is able to deal with previously unseen data. To address the issue that neighborhood contribution may not be the same, veličković et al. [39] proposes Graph Attention Network (GATs) that introduces attention mechanism to neighborhood aggregation. Due to their advantages in dealing with non-euclidean data, GNNs have proved their power in areas like recommender systems [40–42], social network [38] that always make contact with graph-structured data. In rumor detection, Tian et al. [5] proposes Bi-GCN, with top-down direction for rumor propagation and bottom-up direction for rumor dispersion; Yang et al. [8] constructs four types of camouflage strategies which are introduced to GCN through adversarial training; Lin et al. [43] designs a Hierarchical Graph Attention Networks to attend over the responsive posts that can semantically infer the target claim.

2.3 Self-supervised Learning

Deep neural network models can solve supervised learning problems well with enough labeled data. However, manual labels are expensive and may require domain-specific knowledge. Self-supervised learning can generate pseudo-labels, or cluster input through the comparison between positive and negative pairs. Thus studies on self-supervised learning are divided into two branches: *generative models* [13] [44] and *contrastive models* [45].

For generative model variants, one of the most popular models is auto-encoding. This method encodes the input data, and tries to reconstruct it with decoding, during the process noises can be intentionally added to enhance model robustness [13]. Ian et al [46] proposes a generative adversarial network (GAN) for estimating generative models via an adversarial process.

Contrastive models are another important branch in self-supervised learning. Michael et al. [47] defines the notion of Noise Contrastive Estimation (NCE), acting as the objective of most contrastive learning models. The comparison objects can be either context-instance contrast or instance-instance contrast [48]. The former models the relationship between the local part of a sample and its global context representation. Hjelm et al. [45] proposes Deep Infomax (DIM) that maximizes the mutual information between the local part of an image and its global context. Contrastive Predictive Coding (CPC) [49] generalizes this work to speech recognition, which contrasts audio segments with the whole audio. However, Tschannen et al. [50] provides empirical evidence that MI may not be the only reason for this success, and it also strongly depends on the encoding structure and negative sampling strategy. In this case, instance-instance contrastive

focus on instance-level representation instead of MI. One direction [51] [52] [53] iterates between feature encoding and cluster discrimination. The cluster assignment from the discriminator will act as pseudo label signal to improve the quality of the encoder. Another branch is instance-discrimination. He et al. [14] proposes Momentum Contrast (Moco) which substantially increases the number of negative samples. Chen et al. [12] proposes SimCLR that proves the importance of hard positive example, the model follows the end-to-end framework and is trained in quite large batch size. Our method mainly focuses on contrastive learning.

There are also some recent works on SSL for graph data. Veličković et al. [54] Deep Graph Infomax (DGI), which extends DIM to graph-structured data. Kim and Oh [55] proposes superGAT which self-supervise graph attention through edges. Wu et al. [56] applies graph SSL to recommender systems, and presents theoretical analyses of self-supervised graph learning. However, to the best of our knowledge, self-supervised learning has not been explored in the task of rumor detection.

3   Methodology

In this section, we first formulate rumor detection problems (Section 3.1), and introduce 4 core parts of our proposed framework SRD (Section 3.2).

3.1   Task Description

We first introduce some commonly used notations. We would use bold capital letters (e.g. $\boldsymbol{A}$) to represent matrices and lowercase letters (e.g. $\boldsymbol{v}$) to represent vectors. Note that by default all of the vectors are in column, i.e. $\boldsymbol{v} \in \mathbb{R}^d$. X_{ij} stands for the element in matrix X located at row i and column j. Table 1 lists some of the terms mentioned.

As an individual post contains very limited context and is short in nature, we follow previous works [5, 20] and focus on rumor detection at a granular level of event (*i.e.,* a group of posts) instead. Let $\mathcal{C}$ be a set of events, where each event $c \in \mathcal{C}$ originates from a source post with the rumor or truth label y_c. Event c is associated with two heterogeneous data: (1) propagation tree $< \mathcal{V}, \mathcal{E} >$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of post nodes and their connections. In particular, the source post of c serves as the root node, and the other nodes are the replies relevant to the source post. The edge $(v_i, v_j) \in \mathcal{E}$ is directed, representing that post v_i is a reply to v_j. (2) semantic contents, where each post node $v \in \mathcal{V}$ is attached with its content features $\mathbf{x} \in \mathbb{R}^d$, and d is the feature dimension. As a result, we can reorganize each event c in the form of an attributed graph with the adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, whose element is:

$$A_{ij} = \begin{cases} 1, & (v_i, v_j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}. \qquad (1)$$

We formulate the task of rumor detection as follows: given an event c with an attributed graph, we would like to exploit its heterogeneous data (*i.e.,* propagation tree and semantic content), and predict whether the information contained in the event is credible.

3.2   Our Framework

In this section, we present our proposed framework, self-supervised rumor detection (SRD). It is composed of four modules: (1) propagation representation learning, which applies a GNN model on the propagation tree; (2) semantic representation learning, which employs a text CNN model on the post contents; (3) contrastive learning, which models the co-occurring relations among propagation and semantic representations; and (4) rumor prediction,
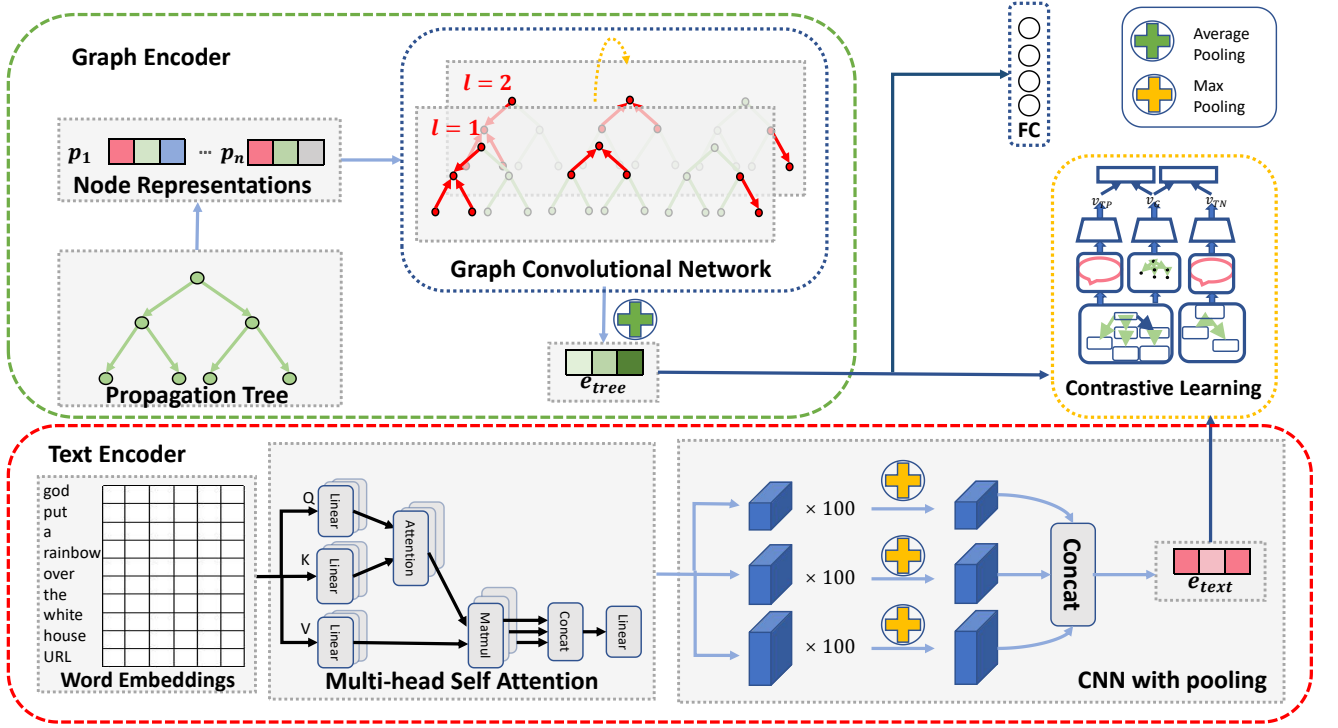
Fig. 2 The framework of our proposed self-supervised learning rumor detection (SRD). The framework consists of 4 modules: Graph Encoder, Text Encoder, Contrastive Learning Module, and FC supervised prediction module. The contrastive learning module here is thumbnail of Fig 3.

Table 1 Notations and Corresponding Explanations

Notation	Explanation
A, D	Adjacency matrix and degree matrix of a graph
$X = [x_1, x_2, \cdots, x_N]^T$	Feature matrix of the attributed graph
W^l, b^l	Trainable weights and bias term at layer l
θ	Model parameters requires gradient
g_i	graph view representation
t_i	text view representation

which builds a predictor model upon the event representations.

3.2.1 Propagation Representation Learning

The propagation tree of a post presents how its information flows along with the source network, especially reflecting the user-user interactions. Prior studies [57] have shown that the propagation patterns of rumors are different from that of the truth. Inspired by this, we use a graph neural network (GNN) to extract such patterns as propagation repre-

sentations of posts. At its core is to apply the neighborhood aggregation scheme on the propagation tree $\mathbf{A}$ of a post, which updates the node representations by aggregating the vectorized information passing from neighbors:

$$\mathbf{H}^{(l)} = \sigma(\mathbf{D}^{-\frac{1}{2}}\hat{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}), \qquad (2)$$

where $\mathbf{H}^{(l)}$ is the node representation at the l-th layer, $\mathbf{H}^{(l-1)}$ is that of previous layer, and $\mathbf{H}^{(0)} = \mathbf{X}$ uses the multi-hot encoding embedding in terms of word distribution in post as the initial representations; $\mathbf{D}$ is the degree matrix, and $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with the identify matrix $\mathbf{I}$ representing self-connections; $\sigma(\cdot)$ is the nonlinear activation function, which we set as ReLU. After adopting L layers recursively, we finalize the representations as $\mathbf{H}^{(L)}$, which encode L-order connectivity among nodes. Here we set L as 2.

Having established the representations of nodes within the propagation tree, we use a readout function to get the representation of the source post. Here we resort to the mean pooling operator $f_{\text{mean-pooling}}(\cdot)$:

$$\mathbf{g} = f_{\text{mean-pooling}}(\mathbf{H}^{(L)}). \qquad (3)$$

3.2.2 Semantic Representation Learning

Most of the time in human society, authorities and experts identify rumors by looking into the textual materials to find suspicious content based on their domain knowledge. Intuitively, semantic features have implications for the credibility of a post. To address this issue, we adopt a multi-head self-attention layer combined with CNN to get a comprehensive representation of the post [8]. Here powerful pre-train models like BERT are not chosen for two reasons: (1) The size of the benchmark datasets are too small to do fine-tuning. (2) They are time-consuming and need a great amount of memory.

Multi-head attention [58] allows for capturing word representations from different subspaces. Considering a multi-head attention layer with h heads, each head is a parallel attention layer. These heads share three input matrices: query matrix $\mathbf{Q}$, key matrix $\mathbf{K}$, and value matrix $\mathbf{V}$. For each head i, these input matrices will be projected into d_k, d_k, d_v dimension subspaces as $\mathbf{Q}_i$, $\mathbf{K}_i$, $\mathbf{V}_i$ through trainable linear projections $\mathbf{W}_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \times d_v}$, respectively. Here we adopt the self-attention scheme where $\mathbf{Q}_i = \mathbf{K}_i = \mathbf{V}_i$. This scheme enables every word to attend to each other, and as a result, every single word will be represented by the most similar words. More formally, at each iteration, vectors in $\mathbf{Q}_i$ serve as *queries* and interact with *keys* in $\mathbf{K}_i$ via inner product scaled by $\sqrt{d_k}$, on which a softmax function is applied to obtain weights for *values* in $\mathbf{V}_i$. We denote the output of the i-th head by $\mathbf{Z}_i$:

$$\mathbf{Z}_i = f_{\text{attention}}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = f_{\text{softmax}}(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}})\mathbf{V}_i. \quad (4)$$

The final output $\mathbf{Z}$ of multi-head attention can be calculated as a linear projection of concatenation of h heads:

$$\mathbf{Z} = f_{\text{multi-head}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = f_{\text{concatenate}}(\mathbf{Z}_1, ..., \mathbf{Z}_h)\mathbf{W}^O, \tag{5}$$

where $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{model}}$.

Multi-head self attention performs extremely well in capturing the long-range dependency. Thus many computer vision tasks, such as image classification [59] and object detection [60], employ self-attention as add-ons to boost the performance of traditional CNNs. Here we borrow this idea to further extract text features. $\mathbf{Z} \in \mathbb{R}^{l \times d_{model}}$ is treated as the input of CNN, where l is the length of the post (padded or truncated if necessary), and d_{model} is the dimension of the representation vector of each word.

A convolution operation is a filter $\mathbf{w} \in R^{h \times d_{model}}$ applied to a window of h words to extract high-level features, where h is called the reception field involving h words simultaneously. For example, feature vector $\mathbf{v}_i$ can be generated from a window of words $\mathbf{z}_{i:i+h-1}$:

$$\mathbf{v}_i = \sigma(\mathbf{w} \cdot \mathbf{z}_{i:i+h-1} + \mathbf{b}), \qquad (6)$$

where $\mathbf{z}_{i:i+h-1}$ is the concatenation of pre-train word representations, *i.e.*, $f_{\text{concatenate}}(z_i, ..., z_{i-h+1})$, $\mathbf{b} \in \mathbb{R}$ is the bias term and σ is the activation function like ReLU. In a sentence, the filter will be applied to every possible window of words:

$$\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{n-h+1}]. \qquad (7)$$

Then the max pooling is adopted over the calculated feature map to highlight the most present feature in the patch:

$$\hat{\mathbf{v}} = f_{\text{max-pooling}}(\mathbf{v}). \qquad (8)$$

Filters with different reception field k focus on words in k-gram pattern. Inspired by [61], we use n filters with various window size, and concatenate the output to serve as the ultimate text representation $\mathbf{t}$:

$$\mathbf{t} = f_{\text{concatenate}}(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_n). \qquad (9)$$

3.2.3 Contrastive Learning

Having derived the propagation representation from the GNN module and the semantic representations from the CNN module, we want to fuse these two heterogeneous representations to uncover their underlying relations and find a more comprehensive portrayal of posts. Although many empirical analyses between content features and spreading patterns are conducted, it is still unclear what features of the content are the critical factors to affect the spreading pattern [62].

From a probability perspective, we suppose text content could provide a low entropy prior for propagation structure identification. Towards this end, we devise the self-supervised learning tasks from the perspectives of instance- and cluster-wise discrimination.

Propagation-Semantic Instance Discrimination (PSID). For a given post, the propagation tree and semantic content simultaneously characterize it from two different views. To model such co-occurring relations, we consider a self-supervised task of post self-discrimination — predicting whether two views are from the same post instance. Formally, we treat the views of the same post instances as the positive pairs, *i.e.,* $\{(\mathbf{g}_i, \mathbf{t}_i)|i \in C\}$, and that of any two different post instances as the negative pairs, *i.e.,* $\{(\mathbf{g}_i, \mathbf{t}_j)|i, j \in C, i \neq j\}$. Note that $\mathbf{g}_i$ and $\mathbf{t}_j$ are generated graph and text representation by corresponding encoder E_1 and E_2, respectively. The positive pairs offer the auxiliary supervision to enforce the consistency between two views of the same posts; whereas, the negative pairs encourage the divergence among different posts. The objective function of post self-discrimination is formulated as:

$$\mathcal{L}_{\text{ssl}} = \sum_{i \in C} -\log[\frac{\exp(s(\mathbf{g}_i, \mathbf{t}_i)/\tau)}{\sum_{j \in C} \exp(s(\mathbf{g}_i, \mathbf{t}_j)/\tau)}] \qquad (10)$$

where $s(\cdot)$ stands for the similarity function, which we set as the inner product; τ is a temperature hyperparameter, which plays a critical role in mining hard negatives.

Propagation-Semantic Cluster Discrimination (PSCD). Going beyond the instance-wise discrimination [12, 14], we are motivated by the prior works [51] and also devise a cluster-wise discrimination task. Since PSID is a contrastive learning method that need to compute pairwise comparisons, its performance relies on the negative sampling strategy. We aim to utilize simple clustering methods like K-means to cluster the data points and enforce consistency between cluster assignments produced from different views (*i.e.,* propagation and semantic) of the same post. Intuitively, similar posts are highly likely to have similar views, thus we can group posts into several clusters and distill cluster-wise features of posts. The self-supervised task can be formulated as maximizing in-group homogeneity. More clearly, the cluster assignment of propagation post would act as pseudo-labels to the semantic representation, and vice versa. Technically, the model consists of two components, encoders E_1 and E_2, which map a post c's propagation representation $\mathbf{g}$ and semantic representation $\mathbf{t}$ into the d-dimensional cluster-wise representations, respectively. Then the discriminator assigns these cluster-wise representations to one

of the K clusters:

$$\min_{\mathbf{S}_1} \sum_{c \in C} \min_{\mathbf{a}_1} \|E_1(\mathbf{g}) - \mathbf{S}_1\mathbf{a}_1\|_2^2 +$$
$$\min_{\mathbf{S}_2} \sum_{c \in C} \min_{\mathbf{a}_2} \|E_2(\mathbf{t}) - \mathbf{S}_2\mathbf{a}_2\|_2^2,$$
$$\text{s.t.} \quad \mathbf{a}_1^\top \mathbf{1} = 1, \quad \mathbf{a}_2^\top \mathbf{1} = 1, \tag{11}$$

where $\mathbf{S}_G$ and $\mathbf{S}_T \in \mathbb{R}^{d \times K}$ separately denote the trainable centroid matrices of graph structure and semantic content, whose k-th column is the centroid representation of the k-th cluster; $\mathbf{a}_1$ and $\mathbf{a}_2 \in \{0, 1\}^K$ are one-hot encoding, which represent the cluster assignments based on graph structure and semantic content, respectively. Here we adopt the iterative optimization framework: given the centroid matrices, we learn the best cluster indicators as $\mathbf{a}_1$ and $\mathbf{a}_2$; then the assignment results guide the optimization of $\mathbf{S}_1$ and $\mathbf{S}_2$. As a result, a set of optimal cluster assignments $\{\mathbf{a}_1^*, \mathbf{a}_2^* | c \in C\}$ are established as the pseudo-labels or supervision signal to enhance the representations of posts, as follows:

$$\mathcal{L}_{\text{ssl}} = \sum_{c \in C} l(f_1(E_1(\mathbf{g})), \mathbf{a}_2) + l(f_2(E_2(\mathbf{t})), \mathbf{a}_1), \tag{12}$$

where $l(\cdot)$ is the negative log-softmax function; $f_1(\cdot)$ is a trainable classifier, which takes the swapped propagation representation $\mathbf{g}$ as input to predict the semantic-aware cluster assignment $\mathbf{a}_2$, analogously to $f_2(\cdot)$. The procedure of PSCD and PSID are shown in Figure 3 and Algorithm 1.

3.2.4 Rumor Prediction

We build a rumor detector model on the top of a post c's propagation representation $\mathbf{g}$:

$$p(c) = \sigma(\mathbf{W}\mathbf{g} + \mathbf{b}), \tag{13}$$

where $p(c)$ presents the probability of the post being the rumor; $\mathbf{W}$ is the trainable matrix, and $\mathbf{b}$ is the

---

Algorithm 1: Training Procedure

Input: Attributed graph with node features, edges and labels $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y}\}$, Number of GNN layers, number of CNN layers, batches and training epochs G_L, C_L, B, E

Output: Classification Result: Rumor or Non-Rumor

1   Initialization: $h_v^0 \leftarrow x_v$;
2   **for** $e=1, ..., E$ **do**
3   **for** $b=1, ..., B$ **do**
4   **for** $l=1, ..., G_L$ **do**
5         $g_v \leftarrow Eq.(2)(3), \forall v \in \mathcal{V}_b$;
         // Propagation representation
6   **for** $l=1, ..., C_L$ **do**
7         $z_x \leftarrow Eq.(4)(5)(6), \forall x \in \mathcal{X}$;
         // Semantic Representation
8   **if** *PSCD* **then**
9   freeze S and get a $\leftarrow Eq.(11)$;
10   freeze a and update S $\leftarrow Eq.(11)$;
       // K-means
11      $\mathcal{L}_{ssl} \leftarrow Eq.(12)$;
12   **if** *PSID* **then**
13   negative sampling;
14      $\mathcal{L}_{ssl} \leftarrow Eq.(10)$;
15     $\mathcal{L}_{main} \leftarrow Eq.(14)$;
16     $\mathcal{L} \leftarrow Eq.(15)$;
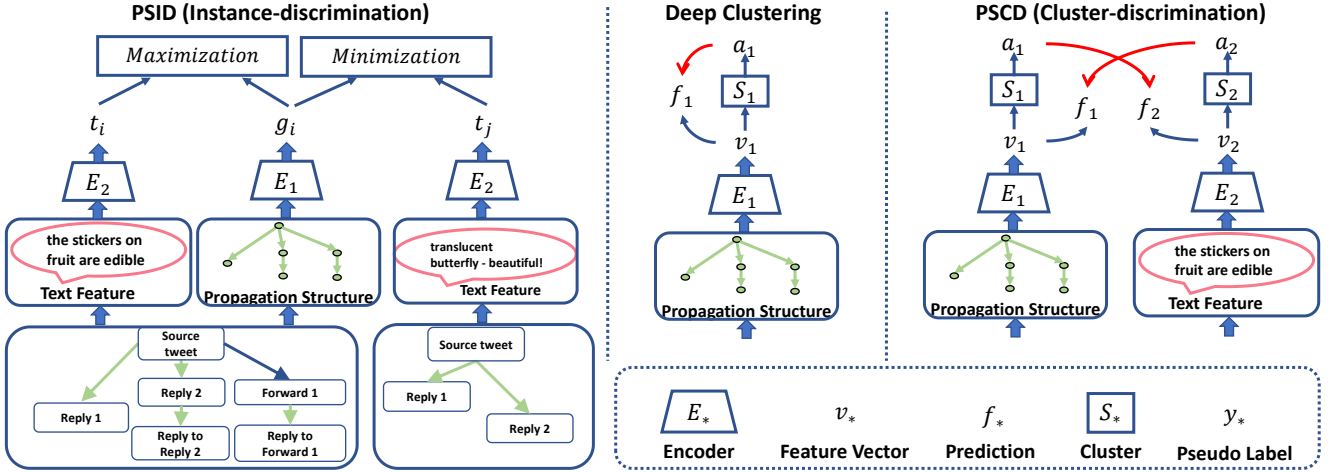
---

bias term. This term is used in a cross-entropy loss, framing the main supervised learning task:

$$\mathcal{L}_{\text{main}} = - \sum_{c \in C} y \log(p(c)), \tag{14}$$

where y is the binary label indicating whether a post is rumor. We adopt a multi-task learning strategy to jointly optimize the main supervised learning task

Fig. 3 Overview of instance discrimination and cluster-discrimination approach. Propagation-Semantic Instance Discrimination (PSID) and Propagation-Semantic Cluster Discrimination (PSCD) are represented. The blue arrows stand for feed forward, while the red arrows stand for back propagation.

and the auxiliary self-supervised learning task:

$$\mathcal{L} = \mathcal{L}_{main} + \lambda \mathcal{L}_{ssl}, \tag{15}$$

where λ is a hyper-parameter to control the strength of the auxiliary task.

4 Experiments and Analyses

We perform experiments on three real-world datasets to evaluate our proposed method, especially the self-supervised learning module. We aim to answer the following research questions:

- RQ1: How does SRD perform as compared with state-of-the-art methods?

- RQ2: How effective are Graph Neural Network, Text Encoder and SSL, respectively, in improving the rumor detection performance of SRD? And how do different hyper-parameter settings (e.g., loss scalar λ, temperature parameter τ) affect our method?

- RQ3: Can SRD perform well on early rumor detection?

4.1 Dataset

We conduct extensive experiments on three widely used benchmark on rumor detection: Twitter [19], Weibo [20], and PHEME [11]. These datasets are representative since they contain blogs and posts from the most influential social media sites in China and United States, respectively. The Weibo dataset contains two binary labels: F stands for False Rumor and T stands for True Rumor, while Twitter dataset and PHEME dataset contain multiple labels: N stands for Non-rumor, F stands for False Rumor, T stands for True Rumor, and U stands for Unverified Rumor. The rumors from Weibo dataset are from the Sina community management center, which reports various misinformation; while non-rumor events are the posts that are not reported as rumors [20]. And the labels in Twitter are converted from binary to quaternary according to the veracity tag of the article in rumor debunking websites (e.g., snopes.com, Emergent.info, etc [19]. The PHEME dataset contains tweets about 9 topics that are closely related to politics and people's livelihood, and the labels are annotated by news practi-

Table 2   Statistics of the Dataset

Statistic	Weibo	Twitter	PHEME
# of posts	3,805,656	204,820	33288
# of users	2,746,818	173,487	-
# of events	4,664	818	6425
# of True Rumors	2,351	205	1067
# of False Rumors	2,313	205	638
# of Unverified Rumors	-	203	698
# of Non-rumors	-	205	4022

tioners. Following [5, 7], we use the same preprocessed dataset: every sample is a propagation tree, whose nodes and edges are posts and their retweet or response relationships. The statistics of the datasets are summarized in Table 2, nonexistent or unknown statistics are marked as "-".

4.2   Baseline

We compared our proposed method with the following strong baselines, including classic and state-of-the-art rumor detection methods.

- **DTC** [15]: A method extracting features manually, based on which they train a Decision Tree Classifier to predict the label of events.
- **SVM-RBF** [16]: SVM Classifier with RBF kernel, using the hand-crafted feature to do classification.
- **SVM-TS** [17]: SVM based model that utilizes time series information to capture temporal features for prediction.
- **SVM-TK** [19]: SVM Classifier takes propagation structure into consideration by designing and leveraging Propagation Tree Kernel.
- **RvNN** [7]: A method that employs Recursive Neural Networks with GRU units, integrates both propagation structure and content semantics for detecting rumors from microblog

posts.

- **STS-NN** [63]: A method that treats the spatial structure and the temporal structure as a whole to model the message propagation.
- **PPC** [27]: A method that focuses on user characteristics to do propagation path classification with a combination of recurrent and convolutional networks.
- **BiGCN** [5]: The first study to solve rumor detection problem with GCN, focusing on both rumor propagation and rumor dispersion with top-down propagation tree and bottom-up propagation tree, respectively.
- **EBGCN** [64]: The first study to explore propagation uncertainty for rumor detection, which adaptively rethinks the reliability of latent relations by adopting a Bayesian approach.

4.2.1   Implementation Details and Metrics

We focus on comparison with the state-of-the-art model, so the other baseline models' experiment results are cited from [5]. We implement BiGCN, EBGCN and our proposed method with Pytorch, and all the three models are tested on Geforce RTX 2080Ti. When running EBGCN on Weibo dataset, it raises an "Out of Memory" issue. To make a fair comparison, we randomly split the dataset into 5 parts with a fixed seed, and employ 5-fold validation to get the robust result. We utilize Adam algorithm [65] as optimizer with default setting unchanged (0.9 and 0.999, respectively). Mini-batch size is set to 128. The learning rate varies during training with cosine annealing strategy [66]. For semantic modules, rare word deletion is adopted to reduce noise in data, *i.e.,* deleting words that appear less than 2 times. Note that this operation could lead to the emptiness of samples, which should also be removed. Text words are truncated or padded to a

fixed length L to form the input matrix. The convolutional kernel size in CNN is set to (3,4,5) with 100 feature maps each, and a dropout rate of 0.5. Word embeddings that act as the input of the text encoder are initialized with 300-dimensional word2vec [67] trained on social media corpora, where words not seen before would be initialized from a uniform distribution. We also try to inference Bert [13] to acquire the representation of sentences, which indicates that the choice of pretrained embedding is not crucial in our method. We keep the word vectors trainable during the process, for propagation information may leverage constraints on them. All of the trainable parameters are initialized with Xavier initializer [68]. For other hyper-parameters, we applied grid search: the start learning rate is chosen between {0.0005, 0.001, 0.01, 0.05}, and the temperature is searched through {0.1, 0.2, 0.5, 1, 2}; λ, which balance the loss of prime learning $\mathcal{L}_{Prime}$ and that of contrastive learning $\mathcal{L}_{cl}$, is tuned within the range of {0.005, 0.01, 0.05, 0.1}.

For fair comparison, We follow the previous work [5, 27] to adopt Accuracy, Precision, Recall and F1 score as indicators to evaluate the overall performance of models.

4.3 Result Analysis

To answer **RQ1**, we compare SRD with the algorithms introduced in Section 4.2, results on Twitter, Weibo and PHEME dataset are shown in Table 3, 4 and 5 respectively. We can observe:

SRD consistently outperforms all of the baselines on all datasets. And as we can see, deep learning models achieve a much better result than models based on hand-crafted features, which is normal because they enable automatic feature engineering and high-level representation. What's more, our proposed method and Bi-GCN perform better than

Table 3 Performance Comparison on Twitter Dataset

Method	Accuracy	N F1	F F1	T F1	U F1
DTC	0.473	0.254	0.080	0.190	0.482
SVM-RBF	0.553	0.670	0.085	0.117	0.361
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.732	0.740	0.709	0.836	0.686
RvNN	0.737	0.662	0.743	0.835	0.708
STS-NN	0.810	0.753	0.766	0.890	0.838
PPC	0.863	0.820	0.898	0.843	0.837
EBGCN	0.871	0.820	0.865	0.922	0.861
Bi-GCN	0.886	0.830	0.881	0.942	0.885
SRD-PSCD	0.891	0.837	0.889	0.945	0.898
SRD-PSID	**0.905**	**0.857**	**0.906**	**0.953**	**0.909**

Table 4 Performance Comparison on Weibo Dataset

Method	Class	Accuracy	Precision	Recall	F1 score
DTC	F	0.831	0.847	0.815	0.831
	T		0.815	0.824	0.819
SVM-RBF	F	0.879	0.777	0.656	0.708
	T		0.579	0.708	0.615
SVM-TS	F	0.885	0.950	0.932	0.938
	T		0.124	0.047	0.059
RvNN	F	0.908	0.912	0.897	0.905
	T		0.904	0.918	0.911
STS-NN	F	0.912	0.912	0.912	0.908
	T		0.911	0.915	0.913
PPC	F	0.916	0.884	0.957	0.919
	T		0.955	0.876	0.913
Bi-GCN	F	0.947	**0.972**	0.921	0.946
	T		0.925	**0.974**	0.949
SRD-PSCD	F	0.949	0.954	0.945	0.949
	T		0.945	0.954	0.950
SRD-PSID	F	**0.962**	0.967	**0.956**	**0.961**
	T		**0.956**	0.967	**0.962**

RvNN and PPC (RNN + CNN), which proves the importance of research in Graph Neural Networks for rumor detection. The state-of-the-art method of rumor detection, the performances of Bi-GCN, EBGCN and STS-NN are less satisfactory com-

Table 5 Performance Comparison on PHEME Dataset

Method	Accuracy	N	F	T	U
		F1	F1	F1	F1
EBGCN	0.819	0.895	0.712	0.702	0.561
Bi-GCN	0.833	0.904	0.731	0.723	0.604
SRD-PSCD	**0.839**	**0.905**	0.721	**0.741**	**0.605**
SRD-PSID	0.838	**0.905**	**0.774**	0.734	0.604

Table 6 Ablation study of SRD demonstrated the advantage of the GCN, text encoder and contrastive learning on Twitter Dataset.

Method	Accuracy	N	F	T	U
		F1	F1	F1	F1
SRD-TEXT	0.803	0.772	0.768	0.885	0.783
SRD-GRAPH	0.880	0.825	0.901	0.924	0.877
Bi-GCN	0.886	0.830	0.881	0.942	0.885
SRD-CONCAT	0.888	0.826	**0.910**	0.940	0.860
SRD-PSID	**0.905**	**0.857**	0.906	**0.953**	**0.909**

pared to our proposed method PSCD and PSID in all of the measurement metrics. Hereby, we attribute our more feasible solution to two following reasons: 1) BiGCN and EBGCN do not incorporate linguistic features or graph structures in post information, instead, they focus more on feature mining in rumor propagation and rumor dispersion; however, the performance of GCN tends to decrease as the number of nodes in propagation tree get smaller because the information provided become sparser. 2) Our proposed method incorporates SSL in the framework, this auxiliary task could improve the performance of the primary task learning, and the text content can provide supplementary information and constraints to the Graph Neural Network branch. Ablation study in Section 4.4 further proves the opinion. The PSID version of SRD performs better than or comparable to PSCD version on three datasets, which demonstrates the effectiveness of introducing negative sampling and pair comparison to boost the performance. The multi-view strategy enables the instance-discrimination approach to benefit from the additional information from the other views, while the cluster-discrimination approach focuses more on the homogeneity of views of the same event.

4.4   Study of SRD (RQ2)

4.4.1   Ablation Analysis

To get deeper insights into the different components (especially the self-supervised learning module) of our method, we fully investigate their impacts.

SRD-TEXT is a variant of SRD, which only uses rumor texts to do prediction.

SRD-GRAPH is a variant of SRD, which only utilizes graph propagation structure and a 2-layer GCN to do prediction.

SRD-CONCAT is a variant of SRD, which incorporates both of the structured information and semantic features in the framework, but directly concatenates them without adding self-supervised learning module.

Table 6 summarizes the comparison results. It can be observed that: The Text-only method is poor performing compared to all the other methods, which shows the importance of modeling the relations between entities, as solely relying on linguistic features to detect rumors can be unstable. BiGCN's performance is better than SRD-GRAPH, the possible reason is that they incorporate the rumor dispersion by adding bottom-up propagation tree. SRD-CONCAT performs better than those methods using either one of text contents or graph feature. This indicates that linguistic features can provide complementary information to propagation patterns, thereby improving the detection results. SRD-PSID outperforms SRD-CONCAT, which supports our opinion that self-supervised learning can better utilize heterogeneous features than simple

Table 7 Effects of the Network Depth

Dataset	Metric	1	2	3	4
Twitter	Accuracy	0.860	0.905	0.879	0.858
	NF1	0.760	0.857	0.818	0.774
	FF1	0.837	0.906	0.882	0.829
	TF1	0.905	0.953	0.921	0.896
	UF1	0.794	0.909	0.896	0.878
Weibo	Accuracy	0.958	0.962	0.955	0.956
	NF1	0.958	0.961	0.944	0.956
	FF1	0.958	0.962	0.958	0.956

Table 8 Effectiveness of propagation structure

Dataset	P Acc	L Acc	% of P and !L	% of L and !P
Twitter	0.898	0.803	11.25	2.50
Weibo	0.954	0.910	8.125	3.75

concatenation. According to Table 8, the column "% of P and !L" is the percentage of the instances which are correctly classified with propagation information rather than linguistic information, analogously to another column. From this table, we can further conclude that the effectiveness of both taking propagation pattern and linguistic features into consideration.

4.4.2 Effects of Network Depth

We vary the network depth L to investigate the efficiency of the usage of multiple embedding propagation layers. In more detail, the layer number is tested within the range of $\{1, 2, 3, 4\}$. We use SRD-1 to represent 1 layer variant, and similar notations for other variants. Figure 7 summarize the result of comparison. We have the following observation:

Clearly, SRD achieves the best result at 2 layers. In the case of network length $L \geq 3$, the model performance begins to decrease, which is consistent with the choice of [5]. We think the possible reason might be that high-order relations between entities,
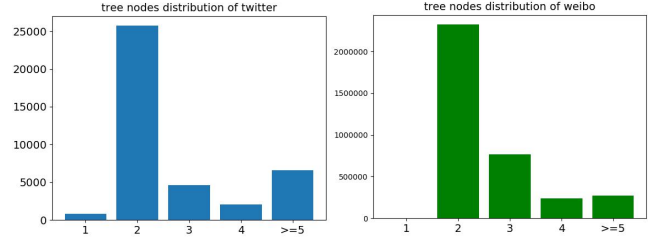

Fig. 4 Depth of Propagation Tree Nodes Distribution on Twitter and Weibo. x axis means the depth of nodes in a propagation tree; e.g. 1 denotes the root node, then 2 will represent the direct children of root node. For better visualization, we classify those whose depth are greater than 5 into a single category.

like third- and fourth-order connectivities, are rare in the dataset; as seen in Figure 4, almost 65 percent of nodes are first-order connections. Thus 3 or more hop neighborhoods contribute little information to the training. Another reason is that Graph Convolution Network suffers more from over-smoothing as layer depth increasing [69]. Analyzing Table 3, Table 4 and Table 7 together, we can find that SRD outperforms baselines which are not based on GCN. It again proves the advantages of taking the rumor propagation pattern into consideration.

4.4.3 Effects of Loss Scalar

As we adopt the *primary and auxiliary* framework, λ should be tuned carefully to avoid the negative influence from the supplementary task when propagating gradient. We search for the near optimal value within a small interval starting from 0. Fig 4 illustrates the performance of SRD with five λ values $\{0, 0.005, 0.01, 0.05, 0.1\}$. As seen from the figure, the performance of SRD changes smoothly near the peak in Weibo dataset and PHEME dataset. It varies more in Twitter dataset, we suppose the possible reason is that the scale of the dataset is relatively small (seen in Table 2). In most cases, SRD is not sensitive to hyperparameter λ. Another
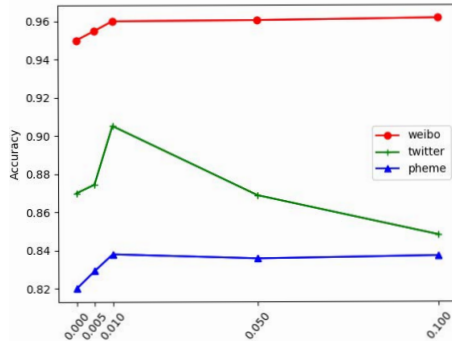
Fig. 5 Model Accuracy **w.r.t** loss scalar λ
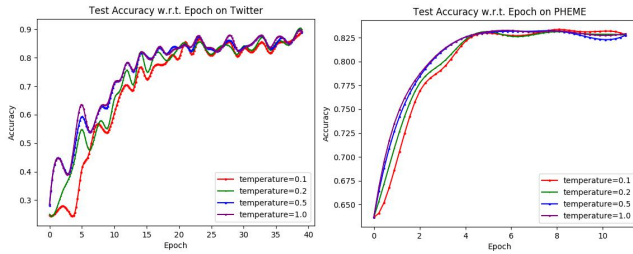
Fig. 6 Model Accuracy **w.r.t** temperature parameter τ

observation is that the performance rises with the increase of the values of λ at the beginning, after reaching the optimal value at around 0.01, it keeps unchanged or decreases in a small value. We also run test with λ equaling to 0.5, only to observe a significant drop in model performance. According to the above observation, we can draw a conclusion that auxiliary task with a small λ can improve the primary task while that with a large one might harm or even mislead the supervised task.

4.4.4 Effects of Temperature Parameter

According to [12,70], temperature parameter τ plays an important role in hard negative sampling and an appropriate τ could benefit the representation learning task when cosine similarity is hired as similarity function. We leverage both *cosine similarity* and *dot product* as our similarity measurement, finding that dot product works better in our task. Fig 6 shows the test accuracy with respect to training epochs on
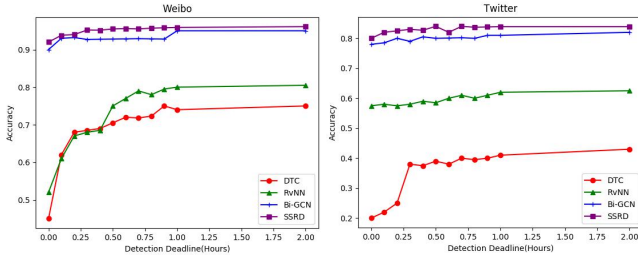
Twitter dataset and PHEME dataset. Weibo dataset converges in three epochs, so it is excluded from the figure. We can observe that: 1) SRD is not sensitive to the value of τ since we adopt dot product rather than cosine similarity, it is consistent with the result in [12]: without l_2 normalization, model with different τ roughly converges to the same point. 2) As the value of τ increasing, the model converges faster, we attribute this phenomenon to the *Uniformity-Tolerance Dilemma* [70]: if the text representation and the propagation pattern from different samples are similar, they are possibly a positive pair but labeled as hard negative mistakenly. This inspires us with the future work of adopting the cosine similarity function with an increasing τ.

4.5 Early Rumor Detection (RQ3)

It is crucial but challenging to detect rumor before it flies high. Many researchers have addressed this issue. Farinneya et al. [71] design an Active Transfer Learning (ATL) strategy to identify rumors with a limited amount of annotated data. Xia et al. [72] propose a state-independent and time-evolving Network which captures the event's unique features in different states. At the early stage, there is limited propagation information to utilize. Thus we expect the text content could help with the issue. To answer **RQ3**, we conduct experiments on early rumor detection. Fig 7 summarizes the comparison result between our proposed method and 3 chosen baselines DTC, RvNN and Bi-GCN. We have the following observation:

For DTC and RvNN, the performance decreases significantly when the detection deadline is restricted to 2 hours, while SRD and Bi-GCN are not affected. Our proposed method is superior to all of the baselines at each deadline, we suppose this gap can be partly attributed to text content representation,

Fig. 7 Early Detection Performance on Weibo Dataset and Twitter Dataset

1	Softmax Score			
	NR	TR	FR	UR
SSRD	1.43e-03	5.16e-03	9.99e-01	6.82e-04
Bi-GCN	0.1464	0.0164	0.8252	0.0119

2	Softmax Score			
	NR	TR	FR	UR
SSRD	5.66e-03	3.50e-03	4.22e-04	9.90e-01
Bi-GCN	0.2858	0.2900	0.1268	0.2974

Fig. 8 Result comparison on two real examples in Twitter. These two examples have the same propagation structure with different ground truth, true rumors and unverified rumors. Softmax scores of prediction results by SRD and Bi-GCN are listed in charts, respectively.

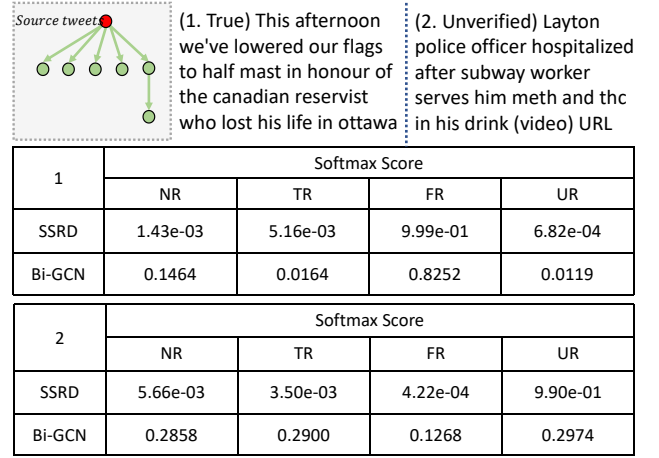which is available once the event begins.

Our proposed method can achieve high performance within half an hour since the release of the source post, which demonstrates that our proposed method can effectively distinguish rumors at a very early stage.

4.6   Case Study

To further validate the importance of textual information, we randomly choose a propagation pattern (e.g. $1 \rightarrow 5 \rightarrow 1$), and see if SRD and BiGCN are able to distinguish between 6 examples sharing this same pattern. Within 6 samples, 2 are true rumors (T) and 4 are unverified rumors (U), note that it's a weak indication that propagation structure to some extent is related to categories of posts. Test runs are performed with SRD and BiGCN, and we summarize the comparison result in Fig 8. There are two key observations:

Both models can distinguish true rumors and unverified rumors, for they can tell that the post is not T/U when the post belongs to U/T by assigning the lowest probability.

SRD predicts the label of a given sample at a quite high confidence level of over 0.99, while the highest softmax score of Bi-GCN, which represents the probability, are 0.8252 and 0.2974 respectively. Especially in the unverified case, Bi-GCN can only tell that this post is not a true rumor, and the scores

of the other 3 classes are similar. From this perspective, SRD is superior to Bi-GCN, which proves the effectiveness of incorporating supplementary information into the primary graph classification task.

5   Conclusion and Future Work

In this work, we proposed a novel method, termed self-supervised learning on rumor detection (SRD), which accounts for both propagation structure and semantic patterns of rumor. By introducing self-supervised learning, we explicitly inject supplementary text representation into the primary rumor detection, which is a supervised graph classification task. In addition, we adopt instance-wise discrimination and cluster-wise discrimination, i.e. NT-Xent loss and XDC to handle the data heterogeneity. Extensive experiments on three real-world datasets demonstrate that SRD performs better than state-of-the-art baselines, hence proving the effectiveness and rationality of incorporating both propagation tree and semantics in the framework. Since our

model utilizes text content, which is available in the early stage of an event, it can detect rumors automatically earlier before state-of-the-art baselines. It is of great necessity since debunking rumors early could avoid the influence from becoming irreversible.

Since the propagation tree is permutation invariant, i.e. children of the same node are treated equally without considering timestamp, in the future, we would further improve SRD by incorporating temporal structures. Another potential direction is to join propagation trees and prepare a big graph for semi-supervised prediction, so that we can generalize SRD from inductive to transductive settings and utilize richer adjacency information. We are also interested in altering GCN encoders. Hyperbolic graph neural network [73] could be an ideal one, for its advantages in dealing with hierarchical social networks. Another substitution is GCN along with adversarial training like CGAT [8], which may be beneficial to model robustness. Also with the development of XAI, we believe graph neural network explainability [74–76] holds great promise in propagation-based rumor detection.

References

1. Farajtabar M, Yang J, Ye X, Xu H, Trivedi R, Khalil E, Li S, Song L, Zha H. Fake news mitigation via point process based intervention. In: Proceedings of the 34th International Conference on Machine Learning, ICML. 2017, 1823–1836

2. Jin Z, Cao J, Guo H, Zhang Y, Wang Y, Luo J. Detection and analysis of 2016 US presidential election related rumors on twitter. In: SBP-BRiMS. 2017, 14–24

3. Ruchansky N, Seo S, Liu Y. Csi: A hybrid deep model for fake news detection. In: Proceedings of the 26th ACM International on Conference on Information and Knowledge Management, CIKM. 2017, 797—-806

4. Shu K, Cui L, Wang S, Lee D, Liu H. defend: Explainable fake news detection. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge, Discovery and Data Mining, KDD. 2019, 395—-405

5. Bian T, Xiao X, Xu T, Zhao P, Huang W, Rong Y, Huang J. Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI. 2020, 549–556

6. Wu Z, Pi D, Chen J, Xie M, Cao J. Rumor detection based on propagation graph neural network with attention mechanism. Expert systems with applications, 2020

7. Ma J, Gao W, Wong K F. Rumor detection on twitter with tree-structured recursive neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL. 2018, 1980–1989

8. Yang X, Lyu Y, Tian T, Liu Y, Liu Y, Zhang X. Rumor detection on social media with graph structured adversarial learning. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI. 2020, 1417–1423

9. Nguyen V H, Sugiyama K, Nakov P, Kan M Y. Fang: Leveraging social context for fake news detection using graph representation. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM. 2020

10. Foroozani A, Ebrahimi M. Anomalous information diffusion in social networks: Twitter and digg. Expert systems with applications, 2019

11. A Z, M L, R P, G W S H, P T. Analysing how people orient to and spread rumours in social media by looking at conversational threads, 2016

12. Chen T, Kornblith S, Norouzi M, Hinton G E. A simple framework for contrastive learning of visual representations. In: ICML. 2020, 1597–1607

13. Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pretraining of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL. 2019, 4171–4186

14. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning, 2020

15. Castillo C, Mendoza M, Poblete B. Information credibility on twitter. In: Proceedings of the 20th International World Wide Web Conference, WWW. 2011, 675–684

16. Yang F, Liu Y, Yu X, Yang M. Automatic detection of rumor on sina weibo categories and subject descriptors. In: Proceedings of the ACM SIGKDD Workshop on

Mining Data Semantics. 2012, 1–7

17. Ma J, Gao W, Wei Z, Lu Y, Wong K F. Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM. 2015, 1751—-1754

18. Liu X, Nourbakhsh A, Li Q, Fang R, Shah S. Real-time rumor debunking on twitter. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM. 2015, 1867–1870

19. Ma J, Wong K F. Detect rumors in microblog posts using propagation structure via kernel learning. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL. 2017, 708—-717

20. Ma J, Gao W, Mitra P, Kwon S, Jansen B J, Wong K F, Cha M. Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI. 2016, 3818—-3824

21. Jin Z, Cao J, Guo H, Zhang Y, Luo J. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In: Proceedings of the 2017 ACM Multimedia Conference, MM. 2017, 795–816

22. Chen T, Wu L, Li X, Zhang J, Yin H, Wang Y. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, 40—-52

23. Guo H, Cao J, Zhang Y, Guo J, Li J. Rumor detection with hierarchical social attention network. In: Proceedings of the 27th ACM international conference on information and knowledge management. 2018, 943–951

24. Li Q, Zhang Q, Si L. Rumor detection by exploiting user credibility information, attention and multi-task learning. In: Proceedings of the 57th annual meeting of the association for computational linguistics. 2019, 1173–1179

25. Yu F, Liu Q, Wu S, Wang L, Tan T. A convolutional approach for misinformation identification. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI. 2017, 3901–3907

26. Kumar S, Asthana R, Upadhyay S, Upreti N, Akbar M. Fake news detection using deep learning models: A novel approach. Transactions on Emerging Telecommunications Technologies, 2020, 1–23

27. Liu Y, Wu Y f B. Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI. 2018, 354–361

28. Rao D, Miao X, Jiang Z, Li R. Stanker: Stacking net-work based on level-grained attention-masked bert for rumor detection on social media. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021, 3347–3363

29. Song Y Z, Chen Y S, Chang Y T, Weng S Y, Shuai H H. Adversary-aware rumor detection. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. 2021, 1371–1382

30. Li J, Ni S, Kao H Y. Meet the truth: Leverage objective facts and subjective views for interpretable rumor detection. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. 2021

31. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations, ICLR. 2017

32. Zhang M, Chen Y. Link prediction based on graph neural networks. In: Proceedings of the 32nd Advances in Neural Information Processing Systems, NIPS. 2018

33. Pan S, Wu J, Zhu X, Zhang C, Philip S Y. Joint structure feature exploration and regularization for multi-task graph classification. IEEE Transactions on Knowledge and Data Engineering, TKDE, 2016, 715–728

34. Wang D, Cui P, Zhu W. Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD. 2016

35. You J, Ying R, Ren X, Hamilton W L, Leskovec J. Graphrnn: Generating realistic graphs with deep autoregressive models. In: Proceedings of the 35th International Conference on Machine Learning, ICML. 2018

36. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu P S. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2021, 4—-24

37. Niepert M, Ahmed M, Kutzkov K. Learning convolutional neural networks for graphs. In: Proceedings of The 33rd International Conference on Machine Learning. 2016, 2014–2023

38. Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs, 2017

39. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representations, ICLR. 2018

40. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR. 2020

41. Wang X, He X, Cao Y, Liu M, Chua T S. Kgat: Knowl-

edge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD. 2019

42. Zeng J, Wang X, Liu J, Chen Y, Liang Z, Chua T, Chua Z L. Shadewatcher: Recommendation-guided cyber threat analysis using system audit records. In: IEEE Symposium on Security and Privacy. 2022

43. Lin H, Ma J, Cheng M, Yang Z, Chen L, Chen G. Rumor detection on twitter with claim-guided hierarchical graph attention networks. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021

44. Oord v. d A, Kalchbrenner N, Espeholt L, kavukcuoglu k, Vinyals O, Graves A. Conditional image generation with pixelcnn decoders. In: Proceedings of the 30th Advances in Neural Information Processing Systems, NIPS. 2016

45. Hjelm R D, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A, Bengio Y. Learning deep representations by mutual information estimation and maximization. In: Proceedings of the 7th International Conference on Learning Representations, ICLR. 2019

46. Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. In: Proceedings of the 28th Advances in Neural Information Processing Systems, NIPS. 2014

47. Gutmann M, Hyvärinen A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, AISTATS. 2010, 297–304

48. Liu X, Zhang F, Hou Z, Wang Z, Mian L, Zhang J, Tang J. Self-supervised learning: Generative or contrastive, 2021

49. Oord v. d A, Li Y, Vinyals O. Representation learning with contrastive predictive coding, 2019

50. Tschannen M, Djolonga J, Rubenstein P K, Gelly S, Lucic M. On mutual information maximization for representation learning. In: Proceedings of the 8th International Conference on Learning Representations, ICLR. 2020

51. Caron M, Bojanowski P, Joulin A, Douze M. Deep clustering for unsupervised learning of visual features. In: the European Conference on Computer Vision, ECCV. 2018

52. Caron M, Misra I, Mairal J, Goyal P, Bojanowski P, Joulin A. Unsupervised learning of visual features by contrasting cluster assignments. In: Proceedings of the 34th Advances in Neural Information Processing

Systems, NIPS. 2020

53. Alwassel H, Mahajan D, Korbar B, Torresani L, Ghanem B, Tran D. Self-supervised learning by cross-modal audio-video clustering. In: Proceedings of the 34th Advances in Neural Information Processing Systems, NIPS. 2020

54. Veličković P, Fedus W, Hamilton W L, Liò P, Bengio Y, Hjelm R D. Deep graph infomax. In: Proceedings of the 7th International Conference on Learning Representations, ICLR. 2019

55. Kim D, Oh A. How to find your friendly neighborhood: Graph attention design with self-supervision. In: Proceedings of the 9th International Conference on Learning Representations, ICLR. 2021

56. Wu J, Wang X, Feng F, He X, Chen L, Lian J, Xie X. Self-supervised graph learning for recommendation, 2020

57. Liu Y, Xu S, Tourassi G. Detecting rumors through modeling information propagation networks in a social media environment. In: Proceedings of 8th International Conference on Social Computing, Behavioral Modeling and Prediction, SBP. 2015

58. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L u, Polosukhin I. Attention is all you need. In: Proceedings of the 31st Advances in Neural Information Processing Systems, NIPS. 2017

59. Bello I, Zoph B, Vaswani A, Shlens J, Le Q V. Attention augmented convolutional networks. In: Proceedings of the International Conference on Computer Vision, ICCV. 2019

60. Wang X, Girshick R, Gupta A, He K. Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR. 2018, 7794—-7803

61. Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014, 1746–1751

62. Zhang Z K, Liu C, Xiu-Xiu Zhan X L, Zhang C X, Zhang Y C. Dynamics of information diffusion and its applications on complex networks. Physics Reports, 2016

63. Huang Q, Zhou C, Wu J, Liu L, Wang B. Deep spatial–temporal structure learning for rumor detection on twitter. Neural Computing and Applications, 2020, 1–11

64. Wei L, Hu D, Zhou W, Yue Z, Hu S. Towards propagation uncertainty: Edge-enhanced bayesian graph convolutional networks for rumor detection. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL. 2021

65. Diederik P. Kingma J B. Adam: A method for stochastic

optimization, 2014

66. Loshchilov I, Hutter F. Sgdr: Stochastic gradient descent with warm restarts. In: Proceedings of the 5th International Conference on Learning Representations, ICLR. 2017

67. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS. 2013, 3111–3119

68. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, AISTATS. 2010, 249—-256

69. Li Q, Han Z, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the 32th AAAI Conference on Artificial Intelligence, AAAI. 2018

70. Wang F, Liu H. Understanding the behaviour of contrastive loss. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR. 2021

71. Farinneya P, Pour M M A, Hamidian S, Diab M. Active learning for rumor identification on social media. In: Findings of the Association for Computational Linguistics: EMNLP 2021. 2021, 4556–4565

72. Xia R, Xuan K, Yu J. A state-independent and time-evolving network for early rumor detection in social media. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020, 9042–9051

73. Chami I, Ying R, Ré C, Leskovec J. Hyperbolic graph convolutional neural networks. In: Proceedings of the 33th Advances in Neural Information Processing Systems, NIPS. 2019

74. Wu Y, Wang X, Zhang A, He X, Chua s T. Discovering invariant rationales for graph neural networks. In: ICLR. 2022

75. Wang X, Wu Y, Zhang A, He X, Chua s T. Towards multi-grained explainability for graph neural networks. In: NeurIPS. 2021

76. Li Y, Wang X, Xiao J, Ji W, Chua s T. Invariant grounding for video question answering. In: CVPR. 2022

Yuan Gao received the M.S. degree in Electrical and Computer Engineering, University of Michigan, Ann Arbor, in 2019. He is now a Ph.D. student in the School of Cyberspace Science and Technology at the University of Science and Technology of China (USTC). His research interest lies in fraud detection, representation learning, and graph learning.

Xiang Wang is now a professor at the University of Science and Technology of China (USTC). He received his Ph.D. degree from National University of Singapore in 2019. His research interests include recommender systems, graph learning, AI explainability, and AI security. He has published some academic papers on international conferences such as NeurIPS, ICLR, KDD, WWW, SIGIR, and AAAI. He serves as a program committee member for several top conferences such as SIGIR and WWW.
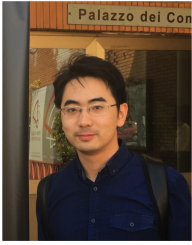
Dr. Xiangnan He is a professor at the University of Science and Technology of China (USTC). He received his Ph.D. in Computer Science from the National University of Singapore (NUS). His research interests span information retrieval, data mining, and multi-media analytics. He has over 80 publications that appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE, TOIS, and TMM. His work has received the Best Paper Award Honorable Mention in WWW 2018 and ACM SIGIR 2016. He is in the editorial board of journals including Frontiers in Big Data, AI Open etc. Moreover, he has served as the PC chair of CCIS 2019 and SPC/PC member for several top conferences including SIGIR, WWW, KDD, MM, WSDM, ICML etc., and the regular reviewer for journals including TKDE, TOIS, TMM, etc.

Huamin Feng is now a professor at Beijing Electronic Science and Technology Institute. He received his Ph.D. dergree from National University of Singapore in 2005. His research interests include multimedia semantic analysis, recommend system, and web content analysis. He has published some academic papers on international conferences such as WWW, SIGIR, and MMM.

Yongdong Zhang (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the University of Science and Technology of China. He has authored more than 100 refereed journal and conference papers. His current research interests include multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology. He was a recipient of the Best Paper Award in PCM2013, ICIMCS 2013, and ICME 2010; and the Best Paper Candidate in ICME 2011. He serves as an Editorial Board Member for Multimedia Systems journal and Neurocomputing.