



## OPEN ACCESS

EDITED BY  
Cong Li,  
Fudan University, China

REVIEWED BY  
Linying Xiang,  
Northeastern University at Qinhuangdao,  
China  
Bo Qu,  
Peng Cheng Laboratory, China  
Yang Chen,  
Fudan University, China

\*CORRESPONDENCE  
Yong Min,  
✉ myong@bnu.edu.cn

SPECIALTY SECTION  
This article was submitted to  
Interdisciplinary Physics,  
a section of the journal  
Frontiers in Physics

RECEIVED 24 November 2022  
ACCEPTED 21 December 2022  
PUBLISHED 06 January 2023

CITATION  
Li Z, Zhang J, Xuan Q, Qiu X and Min Y  
(2023), A novel method detecting  
controversial interaction in the multiplex  
social comment network.  
*Front. Phys.* 10:1107338.  
doi: 10.3389/fphy.2022.1107338

COPYRIGHT  
© 2023 Li, Zhang, Xuan, Qiu and Min. This  
is an open-access article distributed under  
the terms of the [Creative Commons  
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,  
distribution or reproduction in other  
forums is permitted, provided the original  
author(s) and the copyright owner(s) are  
credited and that the original publication in  
this journal is cited, in accordance with  
accepted academic practice. No use,  
distribution or reproduction is permitted  
which does not comply with these terms.

# A novel method detecting controversial interaction in the multiplex social comment network

Zihan Li<sup>1,2</sup>, Jian Zhang<sup>3</sup>, Qi Xuan<sup>1,2</sup>, Xiang Qiu<sup>2</sup> and Yong Min<sup>4,5\*</sup>

<sup>1</sup>Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou, China, <sup>2</sup>The College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, <sup>3</sup>Institute of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, China, <sup>4</sup>The Computational Communication Research Center, Beijing Normal University, Zhuhai, China, <sup>5</sup>School of Journalism and Communication, Beijing Normal University, Beijing, China

With the rise of online social media, users from across the world can participate in opinion formation processes, and some discussions lead to controversial debates around this phenomenon. Controversy detection in social media can help explore public discourse spaces and understand topical issues. Previous controversial detection studies focus more on identifying the opinion or emotional orientation of a comment, while we focus on whether there is a controversial relationship between a comment and its replies. Here, we collect a dataset consisting of 511 news articles, 103,787 comments, and 71,579 users on the Chinese social media platform, Toutiao, and we study the controversial interactions on the subsets of this dataset. Our approach treats news, comments, and users as different types of nodes and constructs multiplex networks connected by user–comment links (i.e., publishing relationship), comment–news links (i.e., comment relationship), and comment–comment links (i.e., replying relationship). Furthermore, we propose a model based on deep learning to detect controversial interactions from these multiplex networks. Our supervised model achieves 83.24% accuracy, with an improvement compared to competitive models. Moreover, we illustrate the applicability of our approach using different ratios of training and testing sets. Our results demonstrate the usefulness of the multiplex networks model for controversial interaction detection and provide a new perspective on controversy detection problems.

## KEYWORDS

**multiplex network, graph embedding, controversy detection, deep learning, graph neural network**

## 1 Introduction

Online social media such as the Chinese platform, Toutiao<sup>1</sup>, have become major channels through which people can easily share their views. In open and free circumstances, the phenomenon of exchanging different opinions probably leads to a fierce discussion and even turns it into a war of words. This situation pollutes the cyber environment. The annoying controversies could be political debates [1, 2] or other topics [3], and the contents of such comments represent a lens of public sentiment. It provides opportunities

1 <https://www.toutiao.com>

to solve the problems in network governance, such as news topic selection, influence assessment, and polarized views alleviation [4].

As a significant part of the procedure, controversy detection in social media has drawn much attention [2, 5]. For example, Twitter is a central place for online discussions and debates. In addition, Twitter provides support for social and political movements. Due to the wide distribution of Twitter, controversial content needs to be censored before it is posted on Twitter. This strategy protects the online environment and limits the spread of misinformation. However, it is challenging to identify controversies in a robust method because of the way controversy varies with the topic of discussion. For example, during the COVID-19 pandemic, wild extremes of emotion and controversial speech have occurred on Twitter. Controversial opinions and subjective judgments are spreading heavily with Twitter retweets [6]. In such cases, controversy detection would be helpful. For example, users will be warned before viewing such posts to prevent them from being influenced by the content of subjective comments in the post [7].

Controversy detection has been studied on web pages and social media for a long decade [8–10]. As for standard web pages, existing methods mainly exploit the controversy detection on Wikipedia [9, 11–13], Reddit [1, 4, 14], and some political blogs [15, 16]. Early methods of controversy detection are mostly based on statistical analysis, such as user edit history in Wikipedia [17, 18], user revision time [13], and context information [12]. Others incorporate the sentiment-based features [19].

Unlike web pages, online social media consist of more diverse topics and fierce discussion among users. There are many controversial comments under a non-controversial topic or post. This makes controversy detection on social media more challenging. The methods during the development of controversy detection can be divided into three types.

**The methods based on the statistics of social media platforms:**

The statistics of the aimed comments, posts, or topics reflect the importance of such items associated with the controversial comment. For example, Koncar et al. [20] extract textural features such as text length and writing style on controversial comments. Moreover, other methods detect controversy standing on leveraging textural and structural features, such as keywords [21], Twitter-specific features [22], and different aspects of discussions including sentiment and topical cohesiveness [23, 24].

**The methods based on the content of comments:** Identifying the semantics of the aimed comment is the directed way to solve controversy detection problems. With the development of natural language processing methods, detecting sentiments based on textural information is more accessible. Based on the data collected from social media platforms, some methods detect controversial snapshots containing several comments attached to the same topic based on the contents of comments and external features [25, 26].

**The methods based on the network structure:** The reply reaction reveals the structural information of social networks. Compared with the traditional homogeneous network, social networks contain different types of entities and relations, rich structural information, and semantic information, which provide a way to discover deeply hidden information for controversy detection. Garimella et al. [2] measured some graph-based features and quantified controversy scores for retweet graphs. In addition, Kumar et al. [27] created a novel LSTM model that combines graph embeddings, user, community, and text features, which gets better prediction results

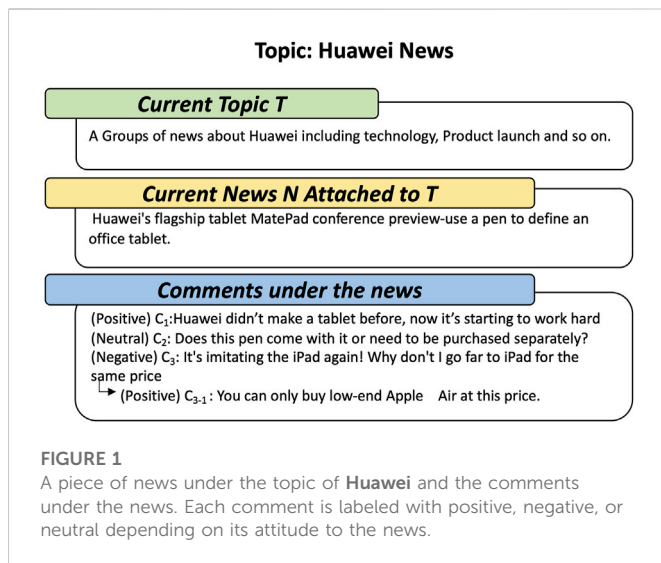
for conflict detection. Hessel et al. [4] integrated the content features with the reply structure and predicted eventual controversy in several Reddit communities. Zhong et al. [14] considered all posts under the same topic using graph-embedding methods. They exploit the information from related posts under the same topic and propose the TPC-GCN and DTPC-GCN models to disentangle both the topic-related and topic-unrelated features and get significant generalizability results on the test datasets.

Graph-embedding techniques have been proven effective in complex network analysis recently. It maps nodes, links, or graphs to low-dimensional dense representations, making it possible to apply machine learning methods for downstream tasks like node classification [28], link prediction [29, 30], and graph classification [31]. Considered as the first work in this area, DeepWalk [32] samples node sequences through random walk and then incorporates skip-gram to learn node embeddings with each node regarded as a word in sentences. To capture both local and global structural features, node2vec [33] extends DeepWalk by employing breadth-first search and depth-first search when sampling node sequences. The sampling strategy is called biased random walk, which improves network representation learning ability. Other random walk-based graph-embedding methods may have similar ideas but adopt different sampling strategies, which focus on different aspects of the network structure [34–36].

Recently, Jiang et al. [37] provided a framework for efficient task-oriented skip-gram-based embeddings. Hu et al. [38] used the generative adversarial networks, which learn node distributions for efficient negative sampling. The emergence of deep learning methods accelerated the growth of this typical research area. Among the variants of graph neural networks (GNN), GCN [39] provided a simplified method to compute the spectral graph convolution to capture the information of the graph structure and node features and transform the structural information between nodes into vectors. GAE [40] designs a decoder for restructuring the relational information of the nodes. The decoder uses the embedding vectors obtained by GCN to reconstruct the graph adjacency matrix, and then perform iteration according to the loss of the reconstructed adjacency matrix and label matrix. In practice, GAE achieves better link prediction results than other algorithms.

Prior work on online controversy detection mainly focuses on comments, topics, or posts [4, 22, 26, 27, 41]. These models are trained to identify the opinion or emotional orientation of a comment, post, or topic. Instead of detecting a controversial comment, our study focuses on the interaction between comments. The controversial comment means this comment probably raises controversy, and the platform's formula determines the score of comments that are likely to be controversial. These definitions may not perfectly characterize whether the current review is prone to controversy because of the limited statics used in the formulation. Controversial interactions are more common than controversial comments in actual social media. In contrast, controversial interaction indicates that a comment's opinion differs from its replies. Compared with quantifying the score of a controversial comment, the controversial interaction can be easily distinguished by the content in the process of labeling.

To detect controversial interactions, we collect information on 103,787 comments and 71,579 users under 511 news and label all of the comments. We treat the users, comments, and news in social media as nodes in multiplex social comment networks. We construct the multiplex networks connected by user–comment links



(i.e., publishing relationship), comment–news links (i.e., comment relationship), and comment–comment links (i.e., replying relationship). Then, we propose the news–comment–user graph convolutional network (NCU-GCN) model for detecting controversial interactions. NCU-GCN integrates the graph structure information and the datasets' dynamic features for obtaining the embedding vector of every comment node. After obtaining the embedding vectors, we classify the interactions on two comments based on the embedding vectors and detect the controversial interactions. Extensive experiments demonstrate that our model outperforms existing methods and can exploit features effectively.

The main contributions of this paper are listed as follows:

1. We build a dataset from a real Chinese news portal platform for controversy detection, which consists of 511 news articles, 103,787 comments, and 71,579 users. This dataset covers a large variety of fields under the topic of **Huawei**, such as finance, technology, and entertainment. Such a real-world dataset can help us better understand the generation of controversy between comments in social media and the evolution of controversy under a comment tree.
2. We propose a GCN-based model, NCU-GCN, for controversy detection on the interaction between two comments. The model can integrate the information from heterogeneous graph structures and dynamic features of edges. The dynamic features of edges mainly consist of the posting time of nodes.
3. Extensive experiments on the Toutiao dataset demonstrate the information of temporal and structural can effectively improve the embedding vectors and get a better result in AUC and AP metrics. Moreover, our model performs generalizability under different ratios of training samples.

The remaining of the paper is organized as follows: first, a detailed description is given for the Toutiao dataset in Section 2 and introduces the proposed NCU-GCN in Section 3. Extensive experiments on Toutiao are presented in Section 4 to show the effectiveness of the proposed methods. Finally, we conclude the paper and highlight some future research directions in Section 5.

## 2 Dataset analysis

In this section, we first give a detailed description of the dataset and data preprocessing, then conduct preliminary descriptive analyses.

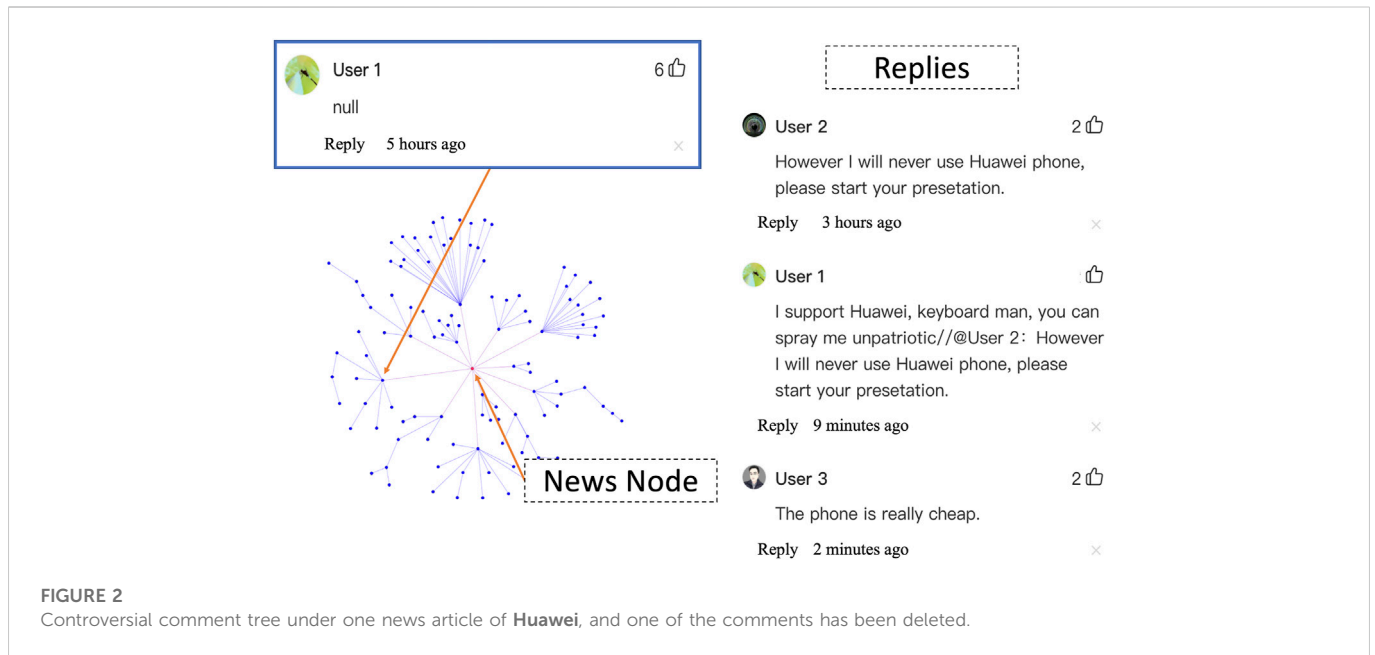
### 2.1 Data description

The data collected from Toutiao include news, users who commented on the news, and the corresponding comments. This news covers a variety of fields, such as technology, finance, and entertainment. In this paper, we focus on **Huawei**-related news, which were released between 2019-03 and 2019-12. Each piece of news is associated with multiple comments, of which the contents present the attitudes of corresponding viewers toward the news. Thus, the comments could be classified into three categories: positive, neutral, and negative. Figure 1 gives an example of controversy over a certain piece of news. News *N* belongs to topic *T*, and it follows multiple comments which show different opinions. As shown in Figure 1, the comments are labeled as positive, neutral, or negative based on their attitudes to the current news, and there exists a situation in  $C_{3-1}$  that expresses refutation literally, but it actually supports *N*. This is because in the comment tree, it refutes to  $C_3$ , a refuting comment to *N*. So, we mark comment  $C_{3-1}$  as a positive comment, and in the progress of data preprocessing, we manually specify the labels of each comment according to its content and context. We collect 511 pieces of news with 103,787 comments in total. Moreover, we label all of the comments with an opinion bias. To make the labeling results more accurate, we started with two annotators labeling the same comment. If a disagreement occurs between two annotators, three additional annotators were added to label this comment. The final labeling results were obtained from the average score of these five annotators. In addition, as Figure 2 indicates, there exist several comments without content and published time. We guess the reason is that the users have deleted these comments. For this kind of comment, we can only judge its label based on its replies.

Given such a large amount of data, we would like to focus on the controversies under *sampled* news in this paper. Thus, we sample three subsets of the data for experiments. Specifically, we first find the top two active users who posted the most comments under different news, and we denote them as  $u_1$  and  $u_2$ . The news commented by  $u_1$ , together with the corresponding comments and the other users, consists of one subset, namely, Toutiao#1. Another subset, Toutiao#2, consists of the news with comments by  $u_2$  and other comments and users, and the common news, which appears in both Toutiao#1 and Toutiao#2, comes to the third subset, namely, Toutiao#3. The basic statistics of the whole dataset and its extracted subsets are presented in Table 1, where interactions mean the relationship between comments. The definition of these interactions is as follows:

- **Controversial interactions:** Positive and negative.
- **Non-controversial interactions:** Positive and positive, neutral and neutral, and negative and negative.
- **Other interactions:** Neutral and positive and neutral and negative.

Also, we provide statistics on the different relationships of the edges in the dataset. Table 2 shows the number of edges between the three types of nodes and the number of edges between the three



**TABLE 1 Basic statistics of the Toutiao dataset and its subsets.**

	Toutiao	Toutiao#1	Toutiao#2	Toutiao#3
#News	511	11	11	1
#Users	71,579	3,496	5,940	1,573
#Comments	103,787	5,570	10,580	2,466
#Controversial interactions	22,184	2,610	4,570	1,166
#Non-controversial interactions	16,339	1,309	2,976	584
#Interactions	46,536	4,995	9,504	2,294

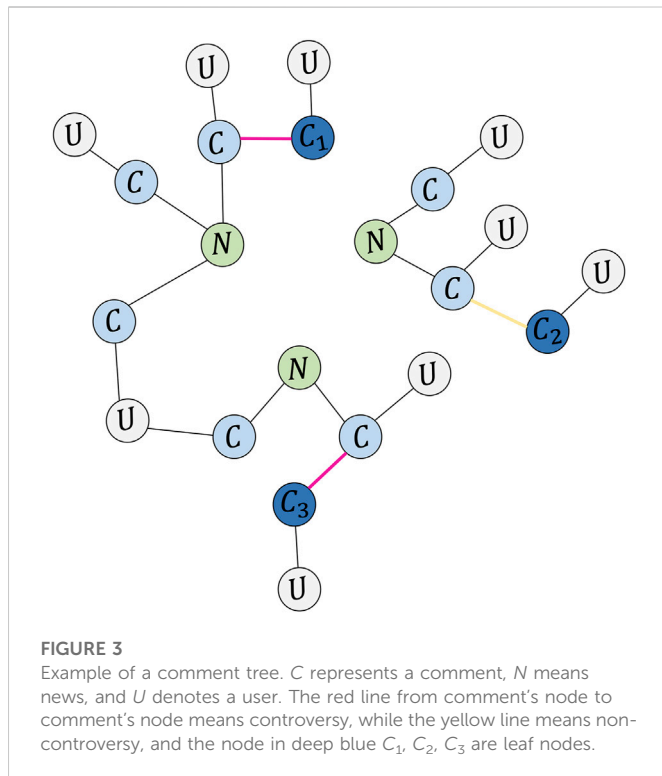
**TABLE 2 Different types of interactions over the Toutiao dataset and its subsets.**

Relationship (A–B)	Toutiao	Toutiao#1	Toutiao#2	Toutiao#3
Comment–user	103,787	4,848	9,105	2,178
Comment–news	57,251	575	1,076	172
Comment–comment	46,536	4,995	9,504	2,294
Positive–negative	22,184	2,610	4,570	1,166
Positive–neutral	4,780	440	792	119
Positive–positive	7,873	253	429	20
Neutral–neutral	5,579	691	2,032	327
Neutral–negative	3,233	636	1,166	425
Negative–negative	2,887	365	515	237

types of nodes with different labels. Comment–user, comment–news, and comment–comment represent the edges between the three different types of nodes in the graph, and the remaining six types represent the edges between the three differently labeled comment nodes.

## 2.2 Analysis from the multiplex social comment network

Despite the content of news and comments, we analyze the contradictory comments from the perspective of the network



structure. Specifically, we construct a multiplex social comment network in which news, comments, and users act as nodes. As Figure 3 shows, these multiplex networks are composed of different entities and relationships. The entities can be classified into the following categories: comment entities, user entities, and news entities. The interactions between comment entities represent the reply relation between these two comments. The interactions between comment entities and user entities mean the related user proposes this comment. The interactions between comment entities and news entities represent the related comment replies to the current news. In the network, the comment nodes are labeled with three different types: positive, negative, and neutral. Regarding the types of these comments, we label the edges connected to two comment nodes into two types: controversial interaction and non-controversial interaction. Controversial interaction implies that the two endpoints of the edges are of positive and negative comments. Non-controversial interaction indicates the two endpoints of the edges are under the same type, which can be both positive and negative. Note that we ignore the links between neutral comments and other kinds of comments in the experiments. On the Toutiao platform, users comment on one or several pieces of news. Consequently, the news is connected to their comments, and comments are linked with their posters; in this way a piece of news and its comments naturally form a comment tree, and multiple comments form a network when they are connected with common users.

As a kind of edge in the network, do controversial edges show any significant structural patterns based on which we could distinguish them from others? This paper adopts the degree and betweenness centrality of two endpoint comment nodes to analyze the controversial edges. The degree of a comment node shows the number of its replies. Betweenness centrality describes the importance of a comment node in the network, which could show whether active yet aggressive users

post the comment. Figure 4 presents the distribution of controversial edges for degree and betweenness centrality. In Toutiao#1 and Toutiao#2, most controversial edges have larger betweenness centrality but a lower degree, and part of them has both relatively large betweenness centrality and degree. Toutiao#3 contains news items in both Toutiao#2 and Toutiao#1. The most controversial edges lie where both betweenness centrality and degree are large. It implies that structural patterns of controversial edges may exist, which motivates us to adopt graph convolution to learn such patterns. We use the time difference of two endpoint comment nodes to analyze controversial edges. Figure 5 presents the distribution of the time difference between the controversial interactions and non-controversial interactions. We observe that the reply time corresponding to controversial edges is shorter than that of non-controversial edges, which is the case in all three datasets.

## 3 Methods

In this section, we first briefly introduce the controversy detection problem and then show how to apply the proposed model, news-comment-user graph convolutional network (NCU-GCN) to solve the problem in detail.

### 3.1 Preliminaries

Our dataset consists of three types of nodes: news, comments, and users. On the platform, users could post their comments to the corresponding news or reply to others' comments, and we construct a news-comment-user network, denoted by  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , to model such relationships.  $\mathcal{V} = \mathcal{V}_n \cup \mathcal{V}_c \cup \mathcal{V}_u$  is the node set for all three types of nodes.  $\mathcal{V}_n$ ,  $\mathcal{V}_c$ , and  $\mathcal{V}_u$  correspond to news, comments, and users, respectively. For a comment  $c$ , it is associated with a timestamp  $t_c$  representing its post time, and each piece of news also has a publishing time  $t_n$ .  $\mathcal{E} = \mathcal{E}_{u,c} \cup \mathcal{E}_{c,c} \cup \mathcal{E}_{c,n}$  represents the three types of relations among news, comments, and users. For  $e_{u,c} = (u, c)$ , where  $u \in \mathcal{V}_u$ , and  $c \in \mathcal{V}_c$ , it denotes that user  $u$  posts comments, and  $e_{c_i,c_j} = (c_i, c_j)$ , where  $c_i, c_j \in \mathcal{V}_c$  represents that comment  $c_i$  replies to comment  $c_j$ . Likewise,  $e_{c,n} = (c, n)$ , where  $n \in \mathcal{V}_n$  shows that comment  $c$  is the first-level comment of news  $n$ . Under the definition, we could obtain an undirected network  $\mathcal{G}$  to model relationships between news, comments, and users.

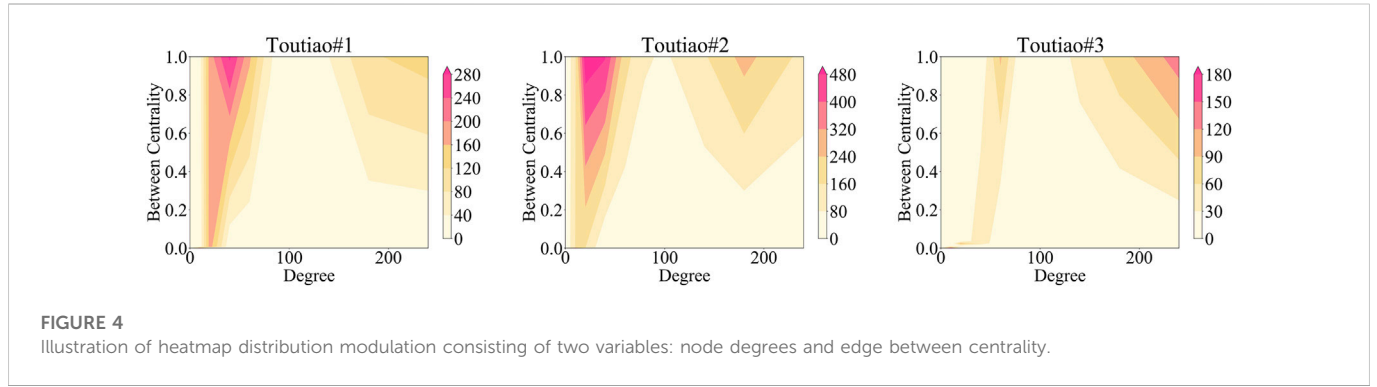
Our solution to find controversy between comments is mainly based on the structure of the news-comment-user network. Mathematically, we aim to learn a mapping  $\mathcal{T}: e_{c_i,c_j} \rightarrow 1$  or  $0$ , where  $1$  means  $c_i$  and  $c_j$  two controversial comments, and  $0$  represents the opposite.

### 3.2 Framework

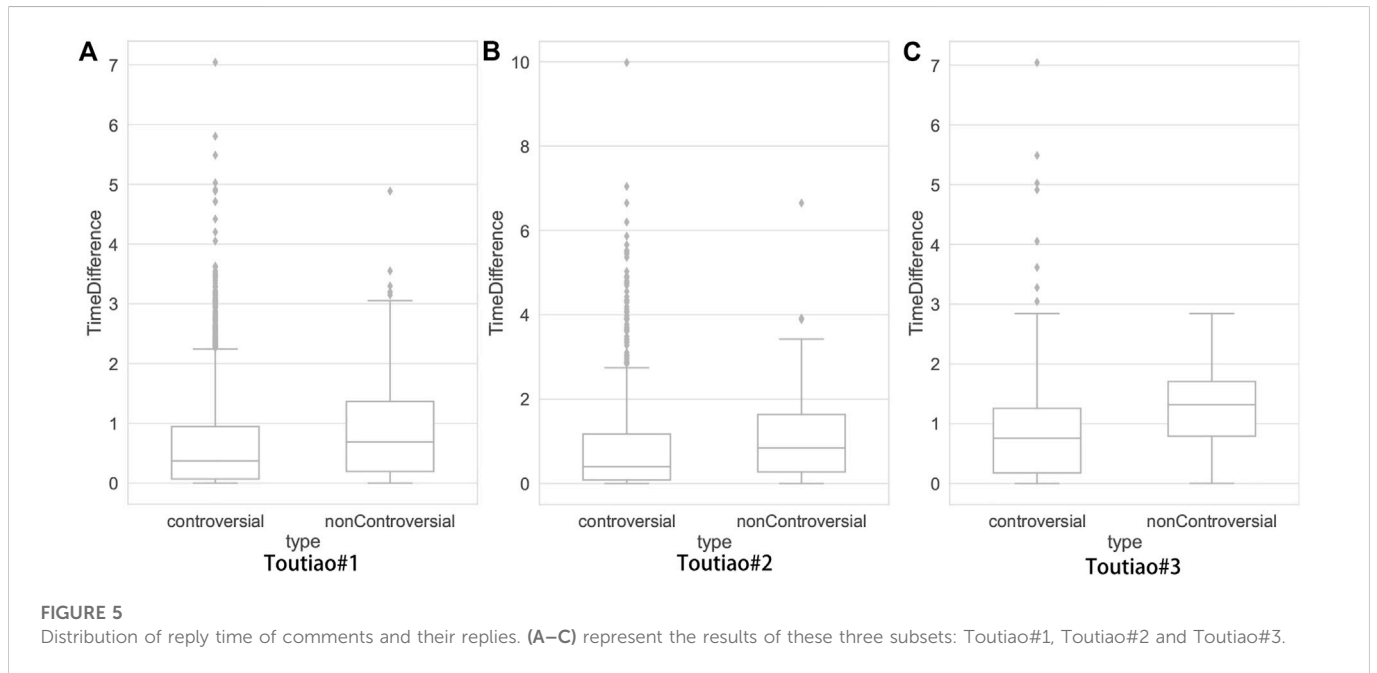
In this part, we give a detailed description of the proposed model NCU-GCN.

GCN has proved its effectiveness in various areas. It aggregates the features of neighboring nodes to learn embeddings for nodes from the local structure perspective. The process is also known as message passing. For node  $i$  in a graph, the hidden representation of  $i$  of  $(l+1)^{th}$  GCN layer, denoted as  $h_i^{(l+1)}$ , could be obtained by





**FIGURE 4** Illustration of heatmap distribution modulation consisting of two variables: node degrees and edge between centrality.



**FIGURE 5** Distribution of reply time of comments and their replies. (A–C) represent the results of these three subsets: Toutiao#1, Toutiao#2 and Toutiao#3.

$$h_i^{(l+1)} = \text{ReLU} \left( \sum_{j \in \mathcal{N}_i \cup \{i\}} W^{(l)} h_j^{(l)} + b^{(l)} \right), \quad (1)$$

where  $\mathcal{N}_i$  is the neighbor of  $i$ .  $W$  is the trainable weights of  $l$ th GCN layer, and  $b^{(l)}$  is the corresponding bias. Incoming messages from  $\mathcal{N}_i$  are aggregated after linear transformation and then passed to the central node. With  $\text{ReLU}(x) = \max(0, x)$  the activation function, a new representation is obtained.

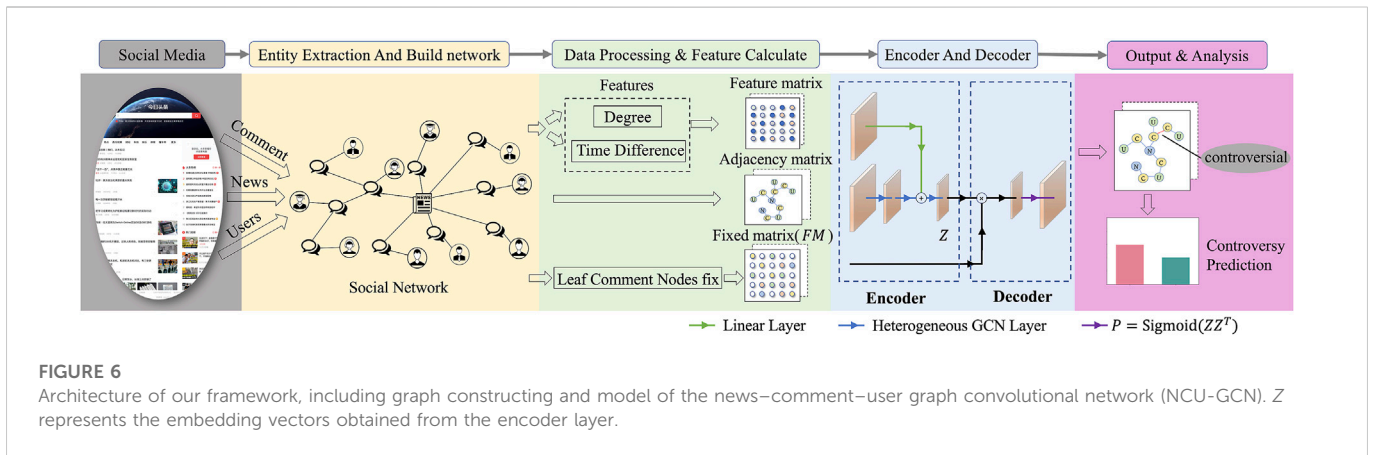
In the message-passing framework, GCN works under the assumption of homogeneity. The comment network we discuss, however, consisted of nodes and edges with multiple types. Thus, we would adapt GCN to the comment network to achieve controversy detection. As presented in Figure 6, the NCU-GCN model is mainly composed of three types of layers: input layer, encoder layer, and decoder layer.

**Input layer:** NCU-GCN receives the comment network as the input. Considering the temporal information between comments and users, we propose to construct a temporal feature matrix  $T \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ , which describes the temporal characteristics in the input layer. Specifically, we consider two types of edges: the edges between comments and the edges between comments and users. The

structural patterns are considered, as shown in Section 2; we define the weight between comment  $i$  and another comment  $j$  by incorporating the degree and the post time interval of the two comments for an arbitrary comment  $i$ . Assuming the comment  $i$  and  $j$  belong to the news  $n$ , we normalize the weight of the two comments by the interval between the post time of the two comments. It is under the assumption that the closer a pair of comments are, the more influence they have on each other. For the relation between  $i$  and its poster, we use a hyper-parameter  $\delta$  to adjust their closeness. For the edges of other types, we set their weights to 0 since they are out of the research scope of this paper. In summary, the construction of  $T$  goes as

$$T_{i,j} = \begin{cases} \frac{\log_{10}(D_i * D_j)}{|t_i - t_j|}, & \text{if } \mathcal{E}_{i,j} \in \mathcal{E}_{c,c}, \\ \delta, & \text{if } \mathcal{E}_{i,j} \in \mathcal{E}_{c,u}, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where  $D_i$  means the degree of comment node  $i$ , and  $t_i$  represents the dynamic features of comment node  $i$ , which is calculated by  $t_i = \log_{10}(\mathbb{T}_i - \mathbb{T}_n + 1) + 1$ ,  $\mathbb{T}_i$  is the post time of comment node  $i$ , and  $\mathbb{T}_n$  is the post time of news which contains comment  $i$ .



**FIGURE 6** Architecture of our framework, including graph constructing and model of the news–comment–user graph convolutional network (NCU-GCN).  $Z$  represents the embedding vectors obtained from the encoder layer.

**Encoder layer:** After obtaining the constructed temporal feature matrix  $T$ , we learn the representations of the nodes, denoted as  $Z \in \mathbb{R}^{|\mathcal{V}| \times d}$ , by a GCN-inspired encoder, and  $d$  is the embedding dimension.  $Z$  is designed to incorporate  $T$ , which describes the temporal characteristics and the structural features learned by neighborhood aggregation. Hence, the encoder consists of GCN-inspired structure learning and temporal feature fusion.

Different from the learning process of GCN, which acts on homogeneous networks, the structure learning module in the encoder layer considers three different kinds of neighbors. As we have introduced, for an arbitrary node in the network, its neighbors could be comment, news, or user nodes. We treat the neighbors separately when learning structural features and then integrate them to obtain the representation. In addition, due to the comment network’s sparsity, we consider all the nodes in a two-hop neighborhood to ensure the abundance of structural features. Mathematically, GCN-inspired structure learning could be described as

$$h_i = \sum_{j \in \mathcal{N}_i^2 \cap \mathcal{V}_n} W_n x_j + \sum_{j \in \mathcal{N}_i^2 \cap \mathcal{V}_c} W_c x_j + \sum_{j \in \mathcal{N}_i^2 \cap \mathcal{V}_u} W_u x_j, \quad (3)$$

where  $h_i \in \mathbb{R}^{|\mathcal{V}| \times d}$  is the representation of node  $i$ .  $W_n$ ,  $W_c$ , and  $W_u$  are the trainable weights for encoding the features of the news, comment, and user node neighbors of  $i$ , respectively. For encoding,  $\mathcal{N}_i^2$  denotes the node set of  $i$ ’s two-hop neighborhood, and thus,  $\mathcal{N}_i^2 \cap \mathcal{V}_n$  is comment node in  $i$ ’s two-hop neighborhood.  $x_i$  is the initial feature of node  $i$ , and we set the initial feature of all nodes as an identity matrix since no other information other than the graph structure is available. Each part in Eq. 3 characterizes corresponding structural features, which performs better than mixing them in a single GCN layer. As for the parts of  $\sum_{j \in \mathcal{N}_i^2 \cap \mathcal{V}_n} W_n x_j$ ,  $\sum_{j \in \mathcal{N}_i^2 \cap \mathcal{V}_c} W_c x_j$ , and  $\sum_{j \in \mathcal{N}_i^2 \cap \mathcal{V}_u} W_u x_j$ , they aggregate the features of news nodes, comment nodes, and user nodes in two-hop neighborhood. The features of different nodes in the graph embedding process of multiplex networks have varying importance to the target node in the feature-aggregation process. In the process, NCU-GCN performs feature extraction by different trainable weight matrices and sums up the computed results. After obtaining the result, we then stack all the embeddings of nodes to obtain the structure feature matrix  $H \in \mathbb{R}^{|\mathcal{V}| \times d}$ :

$$H = [h_0, h_1, \dots, h_{|\mathcal{V}|}]^T. \quad (4)$$

After obtaining the structural feature matrix  $H$ , we further combine the temporal features  $T$  to obtain final representations  $Z \in \mathbb{R}^{|\mathcal{V}| \times d}$ . To achieve the combination, we first map the temporal information with a linear layer to  $\mathbb{R}^{|\mathcal{V}| \times d}$  and then apply the element-wise product over the encoded temporal feature and  $H$ . After the combination, we use a fully connected layer  $f_H$  to smooth the embedding vectors. The process is shown as

$$Z = f_H(\text{concat}(f_L(T), H)). \quad (5)$$

**Decoder layer:** The decoder layer re-transforms the hidden representations of nodes from embedding space to the original network. The edges whose endpoint nodes are close in the embedding space will be reconstructed. In the learning process, the model enlarges the distance of the representations of controversial and non-controversial edges, and they will not be reconstructed when decoding. In this study, we propose to use the inner product as the decoder and the probability matrix  $P = \text{Sigmoid}(ZZ^T)$ . In the comment network, there always exist comments without replies. Such comment nodes carry little structural information, which gets in the way of identifying the relations between them and their parent comments. To cope with the situation, we adopt a `FIXMATRIX`  $F$  in the decoder layer:

$$F_{i,j} = \begin{cases} 1/D_i & \text{if } j \in \mathcal{N}_i, \\ 1 & \text{otherwise,} \end{cases} \quad (6)$$

where  $D_i$  is the out-degree of comment  $i$ ’s parent comment and  $\mathcal{N}_i$  represents the neighbor nodes of the parent node of comment  $i$ . As the definition indicates, the `FIXMATRIX` enhances the representations of leaf comment with other comments being to the same parent comment. We apply a Sigmoid function to the fusion vectors to get the final results:

$$P = \text{Sigmoid}((F^T Z)(F^T Z)^T). \quad (7)$$

$P$  is the possibility matrix, which indicates the prediction result of whether the interaction is controversial or not. To optimize the parameters in the model, we use cross-entropy as the objective:

$$\mathcal{L} = -\frac{1}{|\mathcal{E}'|} \sum_{(i,j) \in \mathcal{E}'} (1 - Y_{i,j}) \log(1 - P_{i,j}) + Y_{i,j} \log(P_{i,j}), \quad (8)$$

where  $\mathcal{E}'$  is the edges used for training.  $Y_{ij}$  is the label with 1 representing controversy and 0 representing the opposite. The framework of NCU-GCN is shown in Algorithm 1.

**Require:** the network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with the adjacency matrix  $A$

- 1: Pretrain the model through deep belief network to obtain the initialized parameters  $\theta = \{\theta^{(1)}, \dots, \theta^{(k)}\}$
- 2: Obtain the dynamic feature matrix  $T$ , apply Eq. 2.
- 3: Obtain FixMatrix  $F$ , apply Eq. 6.
- 4: **while** not convergence **do**
- 5: Based on Eq. 3, obtain structural embedding result  $H$ .
- 6: Based on Eq. 5, use  $T$  and  $H$  to obtain node presentation embedding results  $Z$ .
- 7: Based on Eq. 7, use  $F$  to obtain the probability matrix  $P$ .
- 8: Based on Eq. 8, obtain  $\mathcal{L}$
- 9: Use  $\partial\mathcal{L}/\partial\theta$  to back-propagate through the entire network to get updated parameters  $\theta$ .
- 10: **end while**
- 11: **return**  $P$

**Algorithm 1.** NCU-GCN.

### 3.3 Time complexity

To use NCU-GCN for controversy detection, one has to first calculate the dynamic and structural features and accumulate the result counts to a  $N \times N$  matrix, where  $N$  is the number of nodes. Thus, the time complexity is  $\mathcal{O}(N_{neighbor}N)$ , where  $N_{neighbor}$  is the number of neighbors of one node. The complexity required for a single iteration of NCU-GCN is divided into two main components, encoding operation and decoding operation. The encoder has a time complexity of  $\mathcal{O}(3N^2\mathcal{H}^l) + \mathcal{O}(N^2L)$ , where  $\mathcal{H}$  is the size of the trainable graph filter. We have  $l = 2$ .  $L$  is the output dimension of the linear layer. Assuming that the adjacency matrix is sparse, we can have an encoder with a complexity of  $\mathcal{O}(E\mathcal{H}^2) + \mathcal{O}(EL)$ , where  $E$  is the number of edges from the inputting graph. The complexity of the decoding operation is  $\mathcal{O}((L + \mathcal{H})NF) + \mathcal{O}(3N^2F)$ , where  $F$  is the output dimension of the fully connected layer. Overall, the final time complexity for one epoch is  $\mathcal{O}(E\mathcal{H}^2) + \mathcal{O}(EL) + \mathcal{O}((L + \mathcal{H})NF) + \mathcal{O}(N^2F)$ .

## 4 Results

In this section, we conduct extensive experiments to compare our proposed model with other baselines to show the effectiveness.

### 4.1 Baselines

To validate the effectiveness of our method, we compare NCU-GCN with four representative baselines, which include network embedding approaches and graph neural networks.

- **node2vec** [33] keeps the neighborhood of vertices to learn the vertex representation among networks, and it achieves a balance between homophily and structural equivalence.

- **CTDNE** [42] is a general framework for incorporating temporal information into network embedding methods, which is based on random walk and stipulates that the timestamp of the next edge in the walk must be larger than that of the current edge.
- **GAE** [40] aims at using a structure based on encoder–decoder to get the vertex’s embedding vector of the networks, and then the embedding vectors are used to finish the next purpose. The encoder consists of several GCN layers.
- **metapath2vec** [34] is used for heterogeneous graph embedding based on the meta-path walk to get a series of vertex containing various labels, then heterogeneous skip-gram is used to generate the embedding vectors. The meta-path strategy is set previously, and it is always symmetrical.
- **GAT** [43] achieves advanced performance on graph embedding problems. It uses the attention mechanism to learn the relative weights between two connected nodes. Moreover, GAT further uses multi-head attention to increase the model’s expressive capability

### 4.2 Implementation details

As for baselines, the embedding dimension of node2vec and CTDNE is set to 128, and the optimal key parameters  $p$  and  $q$  of node2vec are obtained through grid search over  $\{0.5, 1, 2\}$ . For the metapath2vec method, the process to classify nodes is as follows: first, we divide all nodes into three types: news, comments, and users, and then we subdivide comment nodes; second, we formulate five meta-paths: including *comments*  $\rightarrow$  *news*  $\rightarrow$  *comments*, *comments*  $\rightarrow$  *comments*, *comments*  $\rightarrow$  *users*  $\rightarrow$  *comments*, *comments*  $\rightarrow$  *news*  $\rightarrow$  *comments*, and *comments*  $\rightarrow$  *users*  $\rightarrow$  *comments*. Based on these meta-paths, we use the metapath2vec method to map the original network to a 128-dimensional vector space for quantization. For all embedding-based methods, we adopt logistic regression (LR) and random forest (RF) as the classifier for controversy detection. In the NCU-GCN model, we use 32-dim latent vectors to represent nodes and we use ReLU for our encoder layer. The optimizer used by NCU-GCN is Adam, and the learning rate is set  $1e - 2$  with a weight decay value, the value is  $1e - 4$ . All the experiments are implemented on the platform equipped with an Intel(R) Xeon(R) Gold 5218 CPU and an NVIDIA Tesla V100S GPU.

Before conducting experiments, we split our dataset into training and testing sets. We divide total edges into two types: controversial and non-controversial. Then, we randomly sample  $n$  edges from the edges labeling controversial and randomly sample the same number of edges from the non-controversial edges. Then, we split into training and testing sets at the ratio of 4:1. Specifically, we choose 80% edges labeling controversial as positive samples of the training set and randomly sample the same number of edges from non-controversial edges. The remaining edges are positive testing data, and the same amount of non-controversial edges are sampled as negative testing data.

### 4.3 Performance comparison

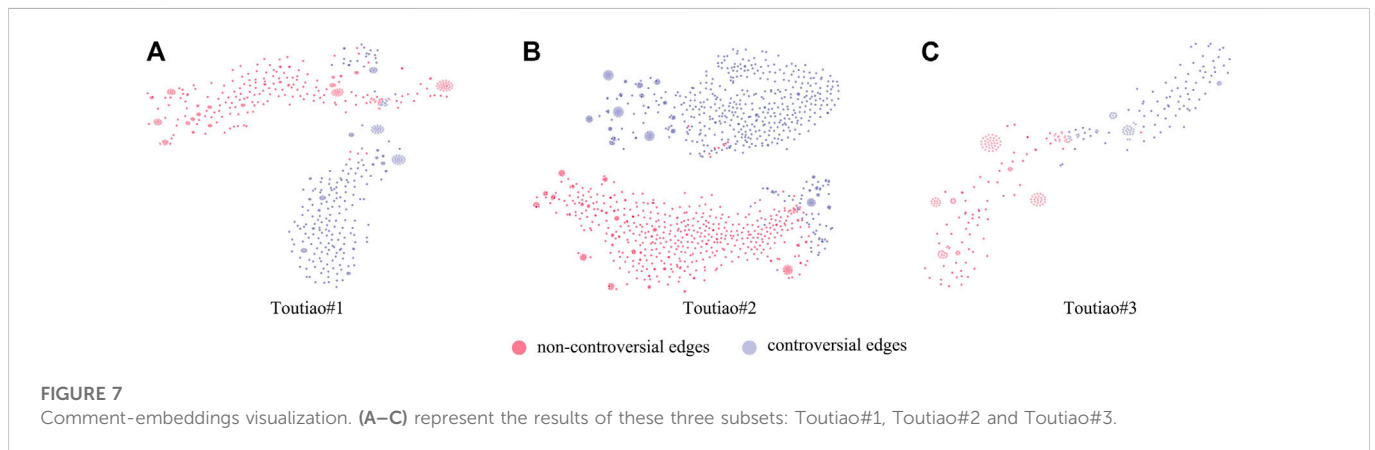
We compare the performance of NCU-GCN with the competitive models on the dataset with AUC and average precision (AP) as the evaluation metrics.



TABLE 3 Performance of different methods among three subsets with respect to AUC and AP.

Method	Classifier	Toutiao#1		Toutiao#2		Toutiao#3	
		AUC	AP	AUC	AP	AUC	AP
node2vec	LR	0.6924	0.6306	0.7266	0.6625	0.7641	0.6973
	RF	0.7559	0.6988	0.8036	0.7386	0.7758	0.7153
CTDNE	LR	0.7066	0.6454	0.7241	0.6589	0.7531	0.6869
	RF	0.7692	0.7044	0.8131	0.7502	0.7878	0.7253
metapath2vec	LR	0.7114	0.6479	0.7191	0.6553	0.7673	0.7032
	RF	0.7621	0.6976	0.8098	0.7396	0.7835	0.7362
GAE		0.7846	0.7694	0.8372	0.8065	0.8051	0.7854
GAT		0.8270	<b>0.8095</b>	0.8490	0.8261	0.8230	0.7992
NCU-GCN(Homo)		0.8215	0.8004	0.8545	0.8201	<b>0.8697</b>	<b>0.8521</b>
NCU-GCN		<b>0.8324</b>	0.8010	<b>0.8671</b>	<b>0.8353</b>	0.8647	0.8457

The bold values stand for the best performance given by the aimed methods.



As presented in Table 3, NCU-GCN outperforms other models on the three Toutiao datasets. As a method only based on random walk, node2vec performs the worst among all methods. CTDNE have better performance than node2vec due to the integration of temporal information when conducting random walk, and metapath2vec makes use of the heterogeneity information in the comment network, which contributes to its better performance compared with node2vec. The performance gap between different classifiers, i.e., LR and RF, indicates that it is also important to choose a proper classifier even for state-of-the-art embedding-based methods. The results of the four GNN (graph neural network) methods show that end-to-end frameworks are always better than making inference by two-step methods, which usually obtain representations first. The NCU-GCN (Homo) method represents a method of graph embedding based on the homogeneity of network nodes, i.e., based on the structure of the NCU-GCN model by modifying Equation (3) as  $\mathbf{h}_i = \sum_{j \in \mathcal{N}_i^c \cap \mathcal{V}} \mathbf{W}_n \mathbf{x}_j$ . Overall, NCU-GCN has better performance than other graph neural network methods, which is due to the consideration of temporal and heterogeneous information when designing the model. Moreover, the results of NCU-GCN are better than NCU-GCN(Homo). This shows that the graph-embedding results are improved by separating the nodes according to their neighborhood node types than by not separating the types.

Moreover, we notice that NCU-GCN performs better on Toutiao#2 than on Toutiao#1 and Toutiao#3. The different expressiveness of the obtained embeddings may cause it. Thus, we visualize the embedding vectors of edges with t-SNE. In order to generate feature representations for an edge, we compose the learned feature representations of the two endpoint comment nodes by element-wise production. As shown in Figure 7, the red points represent non-controversial edges, and the points in blue denote the opposite. Generally, the embeddings of controversial and non-controversial edges could be separated according to the visualizations, and some edges are mixed in the cluster that is not of their type, especially for Toutiao#1 and Toutiao#3. Such edges might result in the decrease of NCU-GCN's performance. The boundary of the two clusters in the visualization of Toutiao#2 is clearer than that in Toutiao#1 and Toutiao#3. It is consistent with the performance recorded in Table 3.

#### 4.4 Parameter sensitivity

Here, we mainly focus on the influence of the three parameters,  $\delta$ ,  $D$ , and  $T$ , on the performance of NCU-GCN. The results are shown in

**TABLE 4 Performance of different parameters of the NCU-GCN method among three subsets with respect to AUC and AP.**

Parameter			Toutiao#1		Toutiao#2		Toutiao#3	
$\delta$	D	T	AUC	AP	AUC	AP	AUC	AP
0	✓	✗	0.8029	0.7343	0.8278	0.7635	0.8569	0.8002
0	✗	✓	0.8347	0.7864	0.8609	0.8264	0.8643	0.8389
0.01	✓	✗	0.8035	0.7331	0.8224	0.7527	<b>0.8659</b>	0.8160
0.01	✗	✓	<b>0.8356</b>	0.7889	0.8623	0.8265	0.8582	0.8456
0.1	✓	✓	0.8284	0.7981	0.8623	0.8265	0.8610	0.8407
0.01	✓	✓	0.8281	<b>0.8071</b>	0.8626	0.8331	0.8626	0.8417
0	✓	✓	0.8324	0.8010	<b>0.8671</b>	<b>0.8353</b>	0.8647	<b>0.8457</b>

The bold values stand for the best performance given by the value of the aimed parameter.

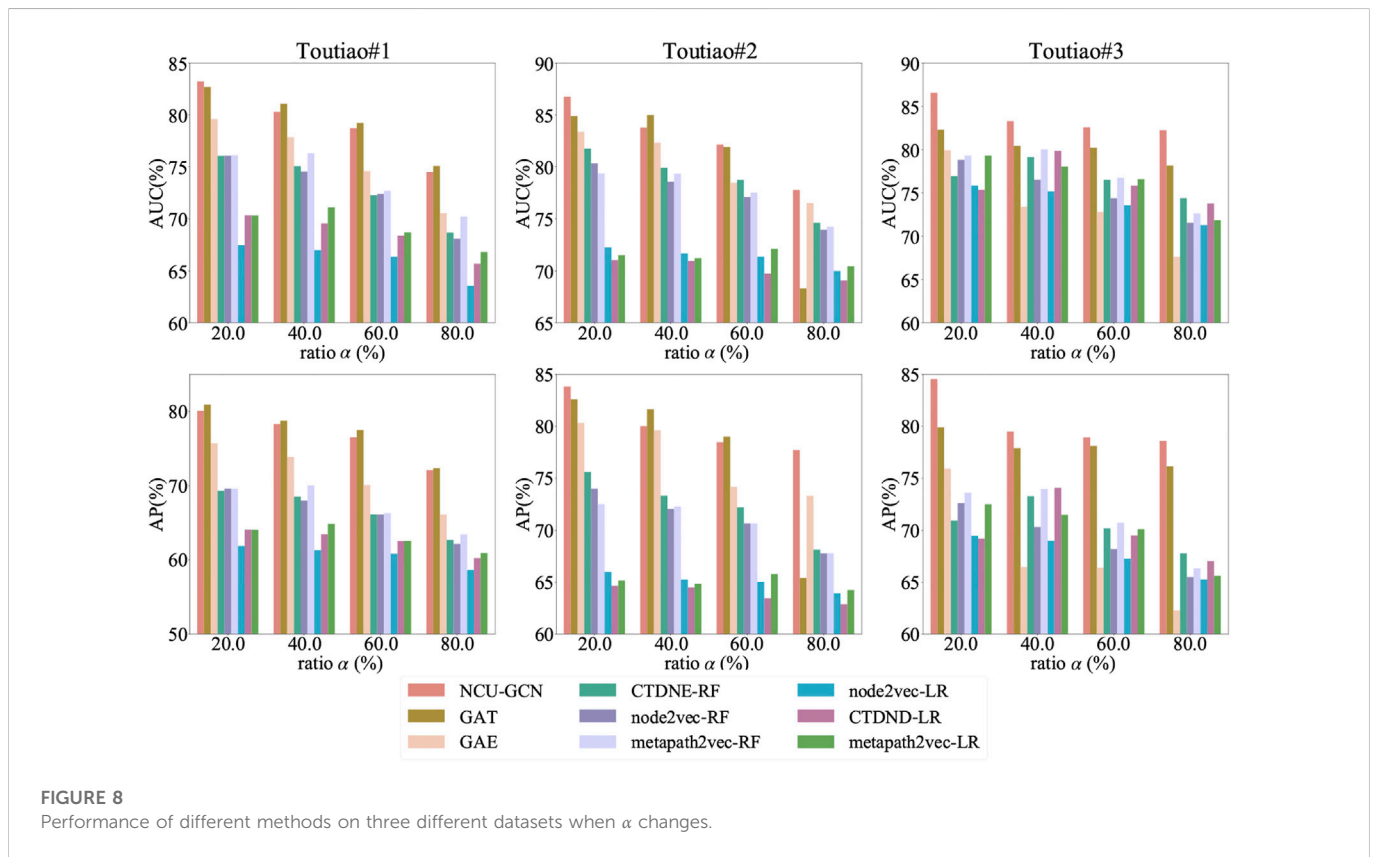
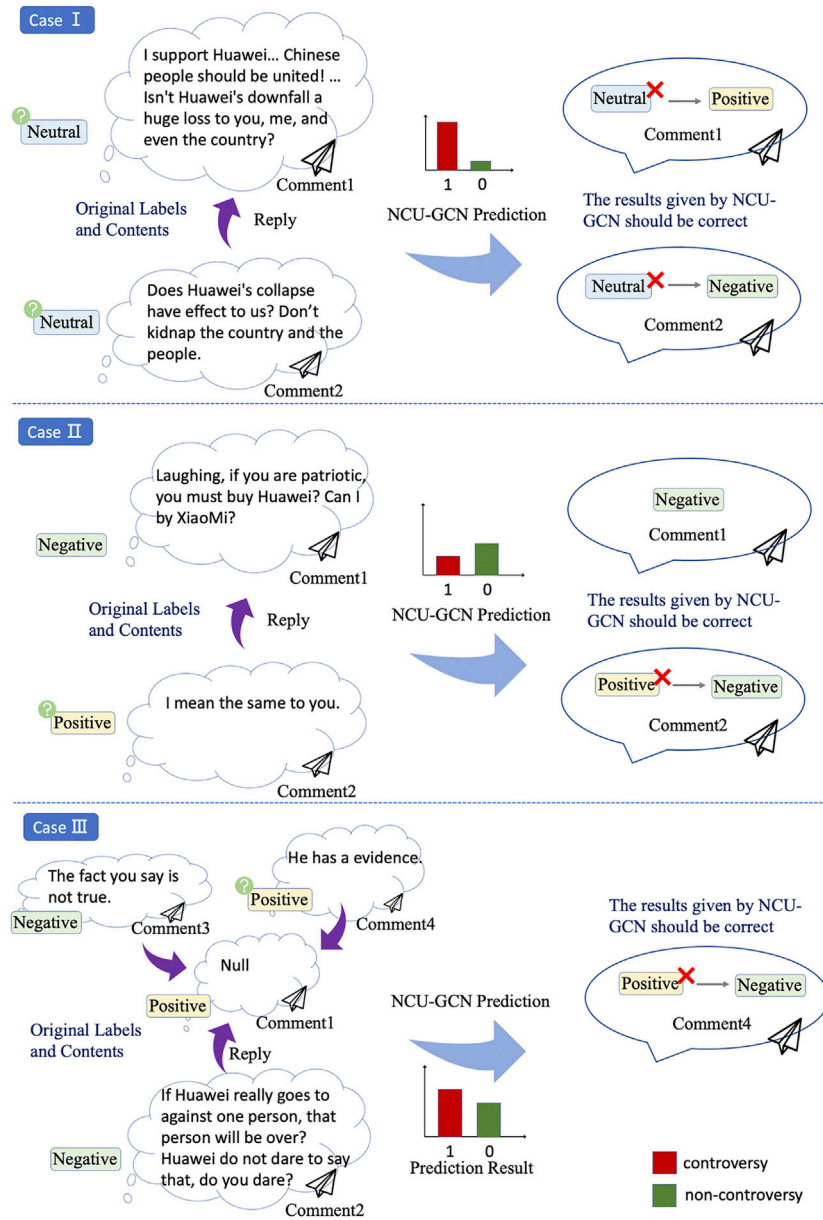


Table 4 where we can see that the performance improves as  $\delta$  decreases. Smaller  $\delta$  represents the shorter time difference between comment nodes and user nodes. It validates that in our Chinese Toutiao datasets, the time difference between comment nodes and user nodes is relatively small. It is also coincidental that there is almost no time difference between the posting and release of a comment on social media. Moreover, the combination of features  $D$  and  $T$  in feature calculation achieves better performance than solely using  $D$  or  $T$ . The larger  $D$  represents that a comment has more replies, which indicates this comment is more important in the network. The results show that the importance of the comment nodes and the reply time difference

can help controversy identification. It is consistent with the conclusions presented in [44].

In addition, we further vary the proportion of test set  $\alpha$  to investigate whether NCU-GCN still performs better when only a small amount of labeled data is available. As shown in Figure 8, the performances of all methods decrease when the amount of training sample reduces. However, NCU-GCN still has the best performance compared to the baselines, showing its superior controversy detection ability. We also notice that the performance of GAE significantly decreases when  $\alpha > 20\%$  on Toutiao#3, and the performance of GAT significantly decreases when  $\alpha = 80\%$  on Toutiao#2. This is because the



**FIGURE 9**  
Samples of the comments which may get a wrong label.

risk of overfitting the model increases as the number of training sets decreases. As a model which also develops from GCN, NCU-GCN adopts the FixMatrix and the temporal feature matrix  $T$  to cope with the problem.

### 4.5 Misjudged cases

In addition to pursuing better performance, we also investigate the false positive edges to find why NCU-GCN fails on these samples. Statistically, a part of error detected edges occur due to the low depth of the endpoint comments in the comment tree. In other words, they possess less structural information than other comments. Although we propose to use FixMatrix to alleviate the

influence, it is still hard to make precise detection under this circumstance.

Also, we find the edges with wrong labels in the misjudged samples, which are more common in real-world applications, especially in semantic-related tasks. As we introduced in Sec. 1, the obscurity of linguistics leads to the difficulty of comment labeling, even for human annotators, and the lack of comment content may also result in wrong labels. Figure 9 shows the three different kinds of misdetection, in which the comments are assigned with wrong labels. In case I, both of the comments are labeled as neutral, and the relation between the two comments is non-controversial consequently, which is opposed to the result given by NCU-GCN. However, the parent's comment is positive and the reply is negative according to their texts, implying that the relationship is controversial. The actual label is

identical to the model's judgment. It is a false-positive case where a non-controversial edge is considered a controversial one by the model. Similarly, case II is a truly negative one under the same situation. In case III, the label of the parent comment is obtained by annotators' inference according to its replies due to the lack of content. It may also lead to wrong comment labels. In these cases, the model makes the right judgments despite the wrong labels. It is because the model learns the structural patterns of controversial comments in the network, again showing the benefits of graph-based learning for controversy detection against the approaches solely based on semantics. It also reveals the dilemma that high-quality annotation is essential but always deficient in real-world applications. We may integrate semantics into the detection model in future works to give human-readable results, which is a possible solution to the aforementioned dilemma.

## 5 Discussion

In this paper, we collect a real-world dataset from a Chinese social media platform and construct multiplex networks connected by multiple links. After obtaining the networks, we propose NCU-GCN to detect controversial interactions between a comment and its replies. Unlike the existing works, we focus on detecting controversial interactions in social comment networks and exploit the information from related news under the same topic and the reply structure for more effective detection. Our paper proposes a novel strategy for multiplex social comment networks based on the graph convolutional network. Through training, NCU-GCN can fully exploit the multiplex social comment network's structural information and dynamic features and gain an improvement compared to competitive models. Moreover, the extensive experiments present NCU-GCN achieving good performance with a low train-test ratio of multiplex social comment networks. Moreover, our method can also find some mislabeled comments. The labels of some comments may be mislabeled, but NCU-GCN can identify them through the graph-embedding method.

This work opens several avenues for future research. First, our solution takes the graph-embedding method to solve the problem of controversial detection. Though some text information is lost, it avoids the problem of inaccurate prediction caused by semantic diversity. Second, we incorporated time information, node importance, and related node information into our model and achieved specific results.

## References

- Coletto M, Garimella K, Gionis A, Lucchese C. Automatic controversy detection in social media: A content-independent motif-based approach. *Online Soc Networks Media* (2017) 3:22–31. doi:10.1016/j.osnem.2017.10.001
- Garimella K, Morales GDF, Gionis A, Mathioudakis M. Quantifying controversy on social media. *ACM Trans Soc Comput* (2018) 1:1–27. doi:10.1145/3140565
- Guerra P, Meira W, Jr, Cardie C, Kleinberg R. A measure of polarization on social media networks based on community boundaries. *Proc Int AAAI Conf Web Soc Media* (2013) 7:215–24. doi:10.1609/icwsm.v7i1.14421
- Hessel J, Lee L. Something's brewing! early prediction of controversy-causing posts from discussion features. In: J Burstein, C Doran, T Solorio, editors. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019; June 2–7, 2019; Minneapolis, MN, USA. Association for Computational Linguistics (2019). p. 1648–59. Long and Short Papers.
- Lu H, Caverlee J, Niu W. Biaswatch: A lightweight system for discovering and tracking topic-sensitive opinion bias in social media. In: Proceedings of the 24th ACM

## Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

## Author contributions

All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by ZL, JZ, and YM. The first draft of the manuscript was written by ZL and checked by JZ, QX, XQ, and YM. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

## Funding

This work was supported in part by the Key Research and Development Programs of Zhejiang Province under Grant 2022C01018, by the National Natural Science Foundation of China under Grants 61973273 and U21B2001, by the Zhejiang Provincial Natural Science Foundation of China under Grants LGF21G010003 and LR19F030001, by the National Social Science Foundation of China under Grant 18BGL231, and by the National Key R&D Program of China under Grant 2021YFB3101305.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

International on Conference on Information and Knowledge Management, Melbourne, VIC, October 19–23, 2015. (2015). p. 213–22.

6. Yin Z, Fan L, Yu H, Gilliland AJ. Using a three-step social media similarity (TSMS) mapping method to analyze controversial speech relating to COVID-19 in twitter collections. In: 2020 IEEE International Conference on Big Data; December 10–13, 2020; Atlanta, GA, USA. IEEE (2020). p. 1949–53.

7. Garimella K, Morales GDF, Gionis A, Mathioudakis M. Reducing controversy by connecting opposing views. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018; July 13–19, 2018; Stockholm, Sweden (2018). p. 5249–53. (ijcai.org).

8. Conover M, Ratkiewicz J, Francisco M, Gonçalves B, Menczer F, Flammini A. Political polarization on twitter. *Proc Int AAAI Conf Web Soc Media* (2011) 5:89–96. doi:10.1609/icwsm.v5i1.14126

9. Dori-Hacohen S, Allan J. Detecting controversy on the web. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, San Francisco, CA, October 28, 2013. (2013). p. 1845–8.

10. Choi Y, Jung Y, Myaeng SH. *Identifying controversial issues and their sub-topics in news articles*. Pacific-Asia Workshop on Intelligence and Security Informatics Springer, Hyderabad, India, June 21, 2010 (2010). p. 140–53.
11. Yasserli T, Sumi R, Rung A, Kornai A, Kertész J. Dynamics of conflicts in wikipedia. *PloS one* (2012) 7:e38869. doi:10.1371/journal.pone.0038869
12. Awadallah R, Ramanath M, Weikum G. Harmony and dissonance: Organizing the people's voices on political controversies. In: Proceedings of the fifth ACM International Conference on Web search and Data mining, Seattle, Washington, February 8–12, 2012. (2012). p. 523–32.
13. Kittur A, Suh B, Pendleton BA, Chi EH. He says, she says: Conflict and coordination in wikipedia. In: Proceedings of the SIGCHI conference on Human factors in computing systems, San Jose CA, April 28–May 3, 2007. (2007). p. 453–62.
14. Zhong L, Cao J, Sheng Q, Guo J, Wang Z. Integrating semantic and structural information with graph convolutional network for controversy detection. In: D Jurafsky, J Chai, N Schluter, JR Tetreault, editors. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Seattle, WA, July 6–10, 2020. Association for Computational Linguistics (2020). p. 515–26. Online, July 5–10, 2020.
15. Mei Q, Ling X, Wondra M, Su H, Zhai C. Topic sentiment mixture: Modeling facets and opinions in weblogs. In: Proceedings of the 16th International Conference on World Wide Web (2007). p. 171–80.
16. Adamic LA, Glance N. The political blogosphere and the 2004 us election: Divided they blog. In: Proceedings of the 3rd International Workshop on Link discovery, Chicago, IL, August 21–25, 2005. (2005). p. 36–43.
17. Vuong BQ, Lim EP, Sun A, Le MT, Lauw HW, Chang K. On ranking controversies in wikipedia: Models and evaluation. In: Proceedings of the 2008 International Conference on Web search and Data mining, Palo Alto CA, February 11–12, 2008 (2008). p. 171–82.
18. Rad HS, Barbosa D. Identifying controversial articles in wikipedia: A comparative study. In: Proceedings of the eighth annual International Symposium on Wikis and Open collaboration, Linz, Austria, August 27–29, 2017 (2012). p. 1–10.
19. Linmans J, van de Velde B, Kanoulas E. Improved and robust controversy detection in general web pages using semantic approaches under large scale conditions. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, October 22–26, 2018 (2018). p. 1647–50.
20. Koncar P, Walk S, Helic D. Analysis and prediction of multilingual controversy on reddit. In: 13th ACM Web Science Conference 2021, Virtual Event United Kingdom, June 21–25, 2021 (2021). p. 215–24.
21. Rethmeier N, Hübner M, Hennig L. Learning comment controversy prediction in web discussions using incidentally supervised multi-task cnns. In: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium, October 31, 2018 (2018). p. 316–21.
22. Addawood A, Rezapour R, Abdar O, Dienes J. Telling apart tweets associated with controversial versus non-controversial topics. In: Proceedings of the Second Workshop on NLP and Computational Social Science, Vancouver, Canada, August 3, 2017 (2017). p. 32–41.
23. Guimarães A, Weikum G. X-posts explained: Analyzing and predicting controversial contributions in thematically diverse reddit forums. In: C Budak, M Cha, D Quercia, L Xie, editors. Proceedings of the Fifteenth International AAAI Conference on Web and Social Media, ICWSM 2021; June 7–10, 2021; Held virtually. AAAI Press (2021). p. 163–72.
24. Levy S, Kraut RE, Yu JA, Altenburger KM, Wang YC. Understanding conflicts in online conversations. In: Proceedings of the ACM Web Conference 2022, Lyon, France, April 25–29, 2022 (2022). p. 2592–602.
25. Popescu AM, Pennacchiotti M. Detecting controversial events from twitter. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, October 26–30, 2010 (2010). p. 1873–6.
26. Choi M, Aiello LM, Varga KZ, Quercia D. Ten social dimensions of conversations and relationships. In: Y Huang, I King, T Liu, M van Steen, editors. WWW '20: The Web Conference 2020; April 20–24, 2020; Taipei, Taiwan. ACM/IW3C2 (2020). p. 1514–25. doi:10.1145/3366423.3380224
27. Kumar S, Hamilton WL, Leskovec J, Jurafsky D. Community interaction and conflict on the web. In: P Champin, F Gandon, M Lalmas, PG Ipeirotis, editors. Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018; April 23–27, 2018; Lyon, France. ACM (2018). p. 933–43. doi:10.1145/3178876.3186141
28. Wang X, Cui P, Wang J, Pei J, Zhu W, Yang S. Community preserving network embedding. Singh S. In: S Markovitch, editor. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence; February 4–9, 2017; San Francisco, California, USA. AAAI Press (2017). p. 203–9.
29. Zhang M, Chen Y. Link prediction based on graph neural networks. In: S Bengio, HM Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, R Garnett, editors. Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018; December 3–8, 2018; Montréal, Canada. NeurIPS (2018). p. 5171–81.
30. Fu C, Zhao M, Fan L, Chen X, Chen J, Wu Z, et al. Link weight prediction using supervised learning methods and its application to yelp layered network. *IEEE Trans Knowledge Data Eng* (2018) 30:1507–18. doi:10.1109/tkde.2018.2801854
31. Lee JB, Rossi R, Kong X. Graph classification using structural attention. In: Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining, London, United Kingdom, August 19–23, 2018. (2018). p. 1666–74.
32. Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, New York, NJ, August 24–27, 2014. (2014). p. 701–10.
33. Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge discovery and data mining, San Francisco CA, August 13–17, 2016 (2016). p. 855–64. doi:10.1145/2939672.2939754
34. Dong Y, Chawla NV, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data mining, Halifax, NS, August 13–17, 2017 (2017). p. 135–44.
35. Wang J, Huang P, Zhao H, Zhang Z, Zhao B, Lee DL. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom, August 19–23, 2018 (2018). p. 839–48.
36. Xuan Q, Wang J, Zhao M, Yuan J, Fu C, Ruan Z, et al. Subgraph networks with application to structural feature space expansion. *IEEE Trans Knowledge Data Eng* (2019) 33:2776–89. doi:10.1109/tkde.2019.2957755
37. Jiang Z, Gao Z, Lan J, Yang H, Lu Y, Liu X. Task-oriented genetic activation for large-scale complex heterogeneous graph embedding. In: Proceedings of The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020 (2020). p. 1581–91.
38. Hu B, Fang Y, Shi C. Adversarial learning on heterogeneous information networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, August 4–8, 2019. (2019). p. 120–9.
39. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017; April 24–26, 2017; Toulon, France. Conference Track Proceedings (2017). (OpenReview.net).
40. Kipf TN, Welling M. *Variational graph auto-encoders* (2016). arXiv preprint arXiv:1611.07308.
41. Yeomans M, Minson J, Collins H, Chen F, Gino F. Conversational receptiveness: Improving engagement with opposing views. *Organizational Behav Hum Decis Process* (2020) 160:131–48. doi:10.1016/j.obhdp.2020.03.011
42. Nguyen GH, Lee JB, Rossi RA, Ahmed NK, Koh E, Kim S. Dynamic network embeddings: From random walks to temporal random walks. In: 2018 IEEE International Conference on Big Data (Big Data); 10–13 December 2018; Seattle, WA, USA. IEEE (2018). p. 1085–92.
43. Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: 6th International Conference on Learning Representations, ICLR 2018; April 30 - May 3, 2018; Vancouver, BC, Canada. Conference Track Proceedings (2018).
44. Coletto M, Garimella K, Gionis A, Lucchese C. A motif-based approach for identifying controversy. *Proc Int AAAI Conf Web Soc Media* (2017) 11:496–9. doi:10.1609/icwsm.v11i1.14949