RD-GCN: A Role-Based Dynamic Graph Convolutional Network for Information Diffusion Prediction

Jingkai Ye, Qing Bao, Ming Xu, Jian Xu, Hongjun Qiu, and Pengfei Jiao

***Abstract*—Information diffusion prediction is an important task which attempts to predict the potential users that will be affected in an information cascade. Existing studies utilize the user interactions in the diffusion graphs with past diffusion records and user relationships in social networks, and capture the structural proximity of those networks to model user similarities for diffusion prediction. However, they ignore structural similarity of the networks, namely user roles, which is of great significance. Actually, a user's future reposting behavior depends on the roles of previous participants. For instance, sometimes people may not respond to discussions of their so-so friends until a more influential friend joins in. Also, user roles change over time. In this paper, we propose a novel Role-based Dynamic Graph Convolutional Network (RD-GCN) which captures the dependencies of dynamic user roles and information diffusion, and jointly learns the two parts. Specifically, the original diffusion graph is divided into subgraphs according to timestamps and users' preferences and roles are captured from those sequential subgraphs and social networks. Also, gated mechanisms are introduced to incorporate users' different tendencies to be influenced by roles. The experiments on three public datasets demonstrate that RD-GCN significantly outperforms state-of-the-art models, verifying the effectiveness of our proposed model.**

***Index Terms*—Information diffusion prediction, social networks, graph neural networks, role-based representations.**

I. Introduction

Nowadays, online social network platforms have become an essential part of our daily life, where we can obtain daily news, repost messages and express personal opinions. These platforms facilitate the generation and diffusion of information, resulting in a large number of information cascades, which provides opportunities for researchers to study how to model and predict information diffusion [1], [2], [3], [4]. The information diffusion prediction task aims to study how information spreads among users in order to predict the future infected users. It has various real-life applications, such as malicious accounts detection [5], [6], rumor detection [7], [8], [9], personalized recommendation [10], online advertising [11], etc.

Early studies of information diffusion prediction apply various predefined diffusion models [12], [13], [14], such as the independent cascade (IC) model and the linear threshold (LT) model [15], and directly learn the diffusion parameters of all social links for diffusion prediction. These models require strict prior assumptions, which limits their model expressiveness. With the recent renaissance of deep learning methods, researchers have proposed a number of deep learning based prediction models. The end-to-end solutions with neural network models can significantly enhance the performance for information diffusion prediction [1], [16], [17]. Existing deep learning based approaches utilize the structure of diffusion graphs and/or social networks for information diffusion. As the diffusion graphs with past diffusion records can reflect users' preferences, most existing works exploit the diffusion graphs for information diffusion prediction [1], [3], [17], [18], [19], [20], [21]. For example, NDM [1] and HiDAN [20] model the diffusion graph structure and capture the interdependencies in the cascades through an attention mechanism. RMTPP [21] unifies the point process and the RNN architecture to model the nonlinear dependencies of past infections, and can output the probability distribution of the next infected user and the corresponding time stamp. MS-HGAT [18] constructs hypergraphs from the diffusion graph to describe the dependencies among users' interactions. The relationships among users in social networks are also considered in previous studies to facilitate information diffusion prediction [4], [16], [22], [23], [24], [25]. The intuition behind is that people have similar interests with their friends, and have a relative higher probability to repost friends' posts. For instance, SNIDSA [4] incorporates the dependencies among users in social networks in the attention mechanism. Deepinf [22] and FOREST [23] use graph neural networks to extract relationships of users' local networks. MC_RGCN [25] proposes a recursive graph convolutional network to learn the higher-order social relationships of users. The above studies have demonstrated the importance to utilize network structure for information diffusion prediction. However, they do not take into account two aspects: First, they only capture the structural proximity of the networks to model user similarities, but do not take into account structural similarity of the networks, which is of great significance for information diffusion. There are cases

Manuscript received 21 December 2023; revised 21 April 2024; accepted 12 May 2024. Date of publication 21 May 2024; date of current version 16 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61806061, Grant 61572165, Grant 61803135, Grant 61702150, Grant 62372146, and Grant 62071327, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LDT23F01012F01, Grant LDT23F01015F01, Grant LQ19F030011, and Grant LTGG23F030004, and in part by the Key Research and Development Plan Project of Zhejiang Province under Grant 2017C01065. Recommended for acceptance by Dr. Zhen Wang. *(Corresponding author: Qing Bao.)*

The authors are with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: qbao@hdu.edu.cn).

Digital Object Identifier 10.1109/TNSE.2024.3403652

where two nodes in a network have the same role or occupy the same position, such as celebrities in social networks, although they do not share a connection or even are far apart, which is called structural similarity or role-based similarity [26]. In fact, the user roles in the networks reflect users' impacts on other users, thus affecting the process of information diffusion. Studies [27] have demonstrated that 25% of the proliferation of information on Twitter is controlled by 1% of the platform's users who act as structural hole spanners, bridging the gap between different communities in the network. Also, about 50% of the URLs on Twitter are generated by 20 K elite users, which account for only 0.05% of the household population [28]. Second, user roles keep changing during the process of information diffusion. For example, an ordinary writer may accumulate experience over time and attract more readers, and eventually become a well-known writer, so his role is changing over time, and the influence on others is also changing. In short, it is necessary to consider dynamic user roles in information diffusion. In recent years, various methods have been proposed for role discovery [26], [29], [30], [31], [32]. For example, ReFeX [29] designs a recursive approach to aggregate nodes' local structure features to obtain global ones, which allows for further extraction of sufficient role features. The features constructed in the ReFeX method are exploited to obtain role-based node representations through feature matrix decomposition by RolX [30]. Recently, the node features are efficiently represented with deep learning methods such as auto-encoders [26], [31]. However, the combination of role-based representation learning and diffusion prediction to predict the next infected users is still unexplored.

To take advantage of the dynamic user roles, we here propose a novel joint learning model named Role-based Dynamic Graph Convolutional Network (RD-GCN). RD-GCN captures the dependencies of dynamic user roles and information diffusion, and jointly learns the two parts. In particular, to dynamically capture users' preferences and roles, we firstly divide the original diffusion graph into subgraphs according to the timestamps. Secondly, to capture users' dynamic preferences, graph neural networks are utilized to separately learn user representations from users' dynamic interactions in the diffusion graphs and users' connections in the social network, which are aggregated with a gated mechanism. Thirdly, to capture users' dynamic roles, a recursive feature extraction method is used to obtain the high-dimensional role features of user nodes for the diffusion graphs of different time intervals, and then multiple deep auto-encoders are applied to encode the features into vector spaces. Furthermore, to account for the different trends of how different people are affected by the role, a gated mechanism is used to aggregate the preference-based and role-based user representations. Additionally, a multi-head self-attention mechanism is utilized to capture the context relevance in the cascades based on the integrated user representations to predict the next infected users. In doing so, the dynamic role-based node representation learning part and diffusion prediction part can be naturally combined and optimized in a unified way.

To sum up, the main contributions of this paper can be summarized as follows:

1) We introduce the notion of structural similarity of the networks, namely user roles, for information diffusion prediction. Also, the users' dynamic roles which change with time are captured. To the best of our knowledge, this is the first work that incorporates the user roles to predict the next infected users in a diffusion cascade.
2) A Role-based Dynamic Graph Convolutional Network (RD-GCN), which captures the dependencies of dynamic user roles and information diffusion process, is proposed for information diffusion prediction. RD-GCN learns user representations of user roles and preferences from sequential diffusion graphs and social networks. To capture each individual user's tendency to be influenced by roles, a gated mechanism is introduced to aggregate representations of user preferences and roles. Finally, the dynamic role-based node representation learning part and diffusion prediction part are optimized in a unified way.
3) Experimental results demonstrate that our model outperforms the state-of-the-art diffusion prediction models on three public datasets, showing the effectiveness and robustness of the model.

The rest of the paper is organized as follows. Section II provides a brief review of related work, and Sections III and IV describe problem formulation and the RD-GCN model in detail. Section V shows the experimental results. Section VI concludes the paper and discusses the future work.

II. Related Work

In this section, we will introduce related work for two major tasks of information diffusion, namely the information popularity prediction and information diffusion prediction. The information popularity prediction task, or called macroscopic diffusion prediction, which predicts diffusion at the macroscopic scale, aims to predict the future sizes of the cascades. The information diffusion prediction task, which predicts diffusion at the microscopic scale, attempts to predict the users to be influenced by certain seed users in an information cascade. This work focuses on the information diffusion prediction task. Also, we will cover the related work of role discovery methods.

Information Popularity Prediction: The information popularity prediction task, which aims to predict the future sizes of the cascades, is an important task for information diffusion. The existing methods of information popularity prediction can be generally classified into three categories, including feature engineering-based methods [33], [34], [35], generative-based methods [36], [37] and deep learning-based methods [2], [38], [39], [40], [41], [42], [43]. Early-stage solutions utilize handcrafted features from feature engineering. For instance, Szabo et al. [33] observe a linear relationship between the future popularity of online content and its early popularity. Later on, Pinto et al. [34] use multiple incremental popularity values to replace the early cumulative popularity. As the structure of communities contains useful information [44], [45], Weng et al. [35] quantify the early diffusion patterns of a meme with the structural characteristics of community concentration to help predict its future popularity. The generative-based methods model the information cascades as event sequences with probabilistic generative methods. For instance, Shen et al. [36] use the reinforced Poisson process (RPP) to simulate three common phenomena in social networks: the attractiveness of a meme, the decay of attractiveness and the rich-get-richer phenomenon. Gao et al. [37] decompose the information cascades with a mixture of sub-processes. With the recent advancements in neural networks, a number of prediction models based on deep learning have emerged. The sequences of infected users are captured for popularity prediction. For instance, DeepCas [38] samples sequences from cascade graphs, which are fed into an RNN model to obtain node embeddings for popularity prediction in an end-to-end approach. LSTMIC [39] uses Long Short-Term Memory (LSTM) to learn sequence patterns from the cascades and predicts whether a cascade will break out. DeepHawkes [2] unifies the Hawkes process and the RNN architecture, and three concepts of Hawkes process are transformed into the DeepHawkes model. Hawkesformer [46] associates hierarchical attention mechanisms with Hawkes self-exciting point processes for popularity prediction. Also, the structural features are incorporated for popularity prediction. CasCN [40] and CoupledGNN [41] utilize graph neural networks to learn the diffusion patterns on the underlying social network. HeDAN [42] adopts heterogeneous graphs to model the nodes and relations of various types, and calculates the node proximities for popularity prediction. MUCas [43] fuses the multi-scale information with multi-scale graph capsule networks and attention mechanisms, and obtains a unique cascade representation to predict the future size of the cascade. CasSeqGCN [47] considers both structural features and temporal features. It utilizes a dynamic routing-based aggregation method to combine node representations learnt via GCN to obtain cascade embeddings in each snapshot, and then employs LSTM to extract temporal information contained in the cascade embeddings for predicting future cascade sizes. Some other related factors are also considered. For example, PEG [48] enhances user preferences to improve popularity prediction performance through three stages: preference topic generation, preference transfer modeling, and social influence activation. PT-GCN [49] employs persistence as a topological data analysis method to extract topological structural features, thereby facilitating the popularity prediction.

Information Diffusion Prediction: The information diffusion prediction task aims to predict the users who will be affected by the seed users in an information cascade. Early studies apply traditional diffusion models [12], [13], [14], such as the independent cascade (IC) model and the linear threshold (LT) model [15], and directly learn the diffusion parameters of all social links for diffusion prediction. The traditional models have been extended to consider dynamic diffusion rates [50], implicit influence [51], time and depth-sensitive influence [51], [52] and etc. These models require strict prior assumptions, which limits their model expressiveness. Recent studies exploit representation learning and deep learning techniques to better capture the temporal dependencies of cascades. Based on the IC model, the Embedded IC model [53] obtains more robust diffusion probabilities by embedding users of social networks into a latent space. Zhang et al. [54] learn community preserving network embeddings from the information cascades with a probabilistic graphical model for better prediction ability. The sequences of infected users are captured with RNN and its variants at the beginning. Among them, TopoLSTM [16] extends the basic LSTM model and utilizes the social graph to model the hidden state relationship as a directed acyclic graph. RMTPP [21] and ERPP [55] unify the point process and the RNN architecture to model the nonlinear dependencies of past infections, and can obtain the probability distribution of the next infected user and the corresponding time stamp. Later on, to better characterize the propagation paths, CYAN-RNN [3] proposes coverage to adjust the misallocation of attention, and the resulting alignments can better reflect the true structure of propagation, HiDAN [20] designs time-aware attention to determine the dependencies of future users on historical users, and NDM [1] alleviates long-term dependence by using convolution neural networks (CNN) to extract the local influence information and encoding the global influence with attention mechanism. MS-HGAT [18] constructs hypergraphs using the diffusion graph to illustrate the dependencies among users' interactions. CE-GCN [19] leverages collaborative patterns in cascades and designs an attention-based cascade aggregator to enhance the learning of user representations for information diffusion. The social network structure is also exploited as people have similar interests with friends. For instance, SNIDSA [4] computes the pairwise similarities of all user pairs based on social network structure and incorporates them into the RNN-based diffusion model. Inf-VAE [56] exploits the social homophily reflected in the social network structure, and proposes a variational auto-encoder framework to jointly model the impact of social homophily and temporal influence. H-Diffu [57] extends Inf-VAE by incorporating latent hyperbolic space to better capture hierarchical structures of social graphs and diffusion cascades separately. MC_RGCN [25] proposes a recurrent graph convolutional network to learn the higher-order social relationship of users for diffusion prediction. In addition, FOREST [23] proposes a reinforcement learning-based multi-scale diffusion prediction model that incorporates macroscopic information like diffusion size into an RNN-based microscopic diffusion model for both macroscopic and microscopic prediction. Some other related factors are also incorporated. For example, MSIDP [58] takes into account specific diffusion timestamps and wide dispersion to capture the users' dynamic retweet preferences, dynamic changes in diffusion rates, and the aggregation of each user's immediate successor nodes. TAN-DRUD [59] captures dual-role user dependencies between information senders and recipients to enhance information diffusion prediction. FADD [60] combines the time attenuation characteristics with a multi-order neighbor influence attenuation mechanism and employs GRU for diffusion prediction.

Role Discovery: There are cases where two nodes in a network have the same role or occupy the same position, such as celebrities in social networks, although they do not share a connection or even are far apart, which is called structural similarity, or role-based similarity [32]. The purpose of role discovery problem is to discern the behavior or function of nodes based on the assumption that nodes with similar roles have similar local connectivity patterns, or higher-order structures. Existing approaches can be classified into three categories: random walk-based approaches, matrix factorization approaches, and deep learning approaches [32]. In the random walk-based approaches, the traditional random walk-based approaches in representation learning are generalized for role discovery. For example, struc2vec [61] uses dynamic time warping distance method to calculate multiple feature-based similarities between all node pairs and performs similarity-based random walks to discover roles. Role2vec [62] introduces the concept of attributed random walks to model the conditional probabilities associated with each role and its context roles. Some approaches utilize structural feature matrix or structural similarity matrix for matrix factorization. For example, ReFeX [29] aggregates local and neighborhood features in a recursive way to obtain global structure information, thus allowing the extraction of sufficient role features. And then, RolX [30] exploits the role features extracted by ReFeX and obtains the role-based node representations through feature matrix factorization. Recently, deep learning approaches are developed. For instance, DRNE [63] uses a layer normalized LSTM with a degree-guided regularizer to learn network embeddings with regular equivalence. DMER [31] uses a graph convolutional network to model the dependencies of nodes from a global perspective, while reconstructing role features with an auto-encoder to obtain role-based representations, and the two parts are integrated for mutual enhancement. The node role features are represented with deep auto-encoders along with an attention mechanism to model node dependencies in RDAA [26]. RDNE [64] learns multiple embeddings for individual users by considering their temporal structural features, and utilizes GRU to capture the relationships between a user's historical and current information.

III. Problem Formulation

A given social network that describes the underlying relationships among users on the social platform is denoted as $G_F = (U, E)$, where U is the node set representing the users in the social network, E is the edge set, and $(u_i, u_j) \in E$ represents a friendship relationship between user u_i and user u_j. Also, the users' reposting behaviors observed on the social network is denoted as a cascade set C, where the m-th cascade $c^m \in C$ is represented as a sequence of k users, i.e., $c^m = \{(u_1^m, t_1^m), (u_2^m, t_2^m), (u_3^m, t_3^m), \ldots, (u_k^m, t_k^m)\}, u_i^m \in U$. Here, (u_i^m, t_i^m) indicates that user u_i^m reposts a message at a specific timestamp t_i^m. Users in the cascades are sorted by time, thus $t_{i-1} < t_i$. Based on the cascades C, a set of diffusion graphs, denoted as G_D, can be constructed to record the dynamic message reposting relationships among users. Instead of using a global diffusion graph to record all the reposting relationships, we construct a set of diffusion graphs which are subgraphs of the global diffusion graph to preserve the dynamic interactions. In particular, the entire historical information diffusion time period $[t_{\min}, t_{\max}]$ is divided into T time intervals, and the cascade set C is used to construct T diffusion graphs, i.e., $G_D = \{G_D^1, G_D^2, \ldots, G_D^T\}$, for the corresponding time intervals. As shown in Fig. 1, four cascades are observed, where nodes represent users and edges represent reposting relationships. The entire historical time period is divided into three time intervals, and the edges of different colors depict reposting relationships occurring at different time intervals. Then, three diffusion graphs are constructed for the corresponding time intervals. Each diffusion graph contains the edges involved in the current as well as previous time intervals. Based on the observed social network and cascades, the task of information diffusion prediction is to predict the next user u_{k+1}^m who will repost the message given the new sequence c^m.

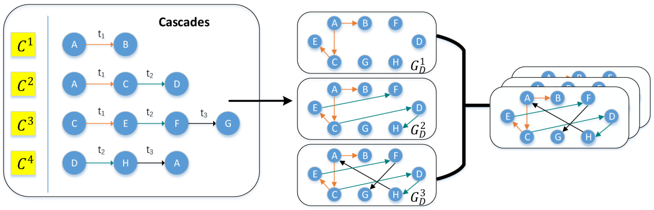


Fig. 1. The construction of diffusion graphs for different time intervals. Four cascades are observed, and c^m represents the m-th diffusion cascade, where nodes represent users and edges represent reposting relationships. The entire historical time period is divided into three time intervals, and the edges of different colors represent the reposting relationships at different time intervals. Finally, three diffusion graphs are constructed, and each diffusion graph contains the edges involved in the current as well as previous time intervals.

IV. The Proposed Model

In this section, we put forward our proposed model, the Role-based Dynamic Graph Convolutional Network (RD-GCN). The framework of RD-GCN is shown in Fig. 2, which contains five parts: 1) Dynamic role-based representation learning module: The high-dimensional dynamic role features are extracted by the recursive feature extraction algorithm. Then, the role features are encoded into low-dimensional role-based node representations through multiple auto-encoders. 2) Representation learning from users' interactions module: The representations of user nodes are learnt from the static social network and the dynamic diffusion graphs through multiple graph convolutional networks. 3) Gated fusion module: the node representations learnt from static and dynamic interactions are first integrated through a gated mechanism to incorporate the importance of different types of interactions. The integrated node representations and role-based node representations are further integrated to incorporate each individual user's tendency to be influenced by roles through another gated mechanism. 4) Prediction module: A multi-head self-attention module is used to capture the context-dependency within each cascade and perform diffusion prediction. 5) Joint learning module: The dynamic role-based node representations and information diffusion are jointly learnt.

A. Dynamic Role-Based Representation Learning

The user roles in the social network reflect their own impacts on other users, affecting the process of information diffusion. Thus, modeling the role information can help diffusion prediction. Based on our understanding, user nodes with the same role in the social network are not necessarily closely

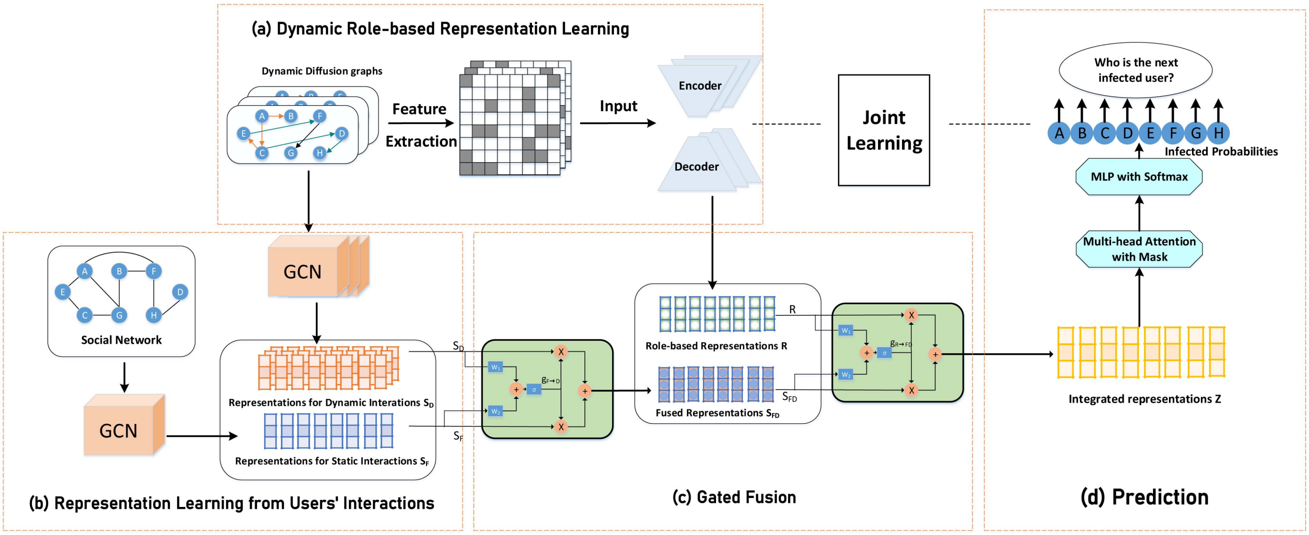

(a) Dynamic Role-based Representation Learning
Dynamic Diffusion graphs
Feature
Extraction
Input
Encoder
Decoder
Joint
Learning
Who is the next infected user?
A B C D E F G H
Infected Probabilities
MLP with Softmax
Multi-head Attention with Mask
Integrated representations Z
(d) Prediction
Social Network
GCN
GCN
Representations for Dynamic Interations S_D
Representations for Static Interactions S_F
(b) Representation Learning from Users' Interactions
S_D
S_F
W_1
W_2
$g_{F\to D}$
σ
Role-based Representations R
R
Fused Representations S_{FD}
S_{FD}
$g_{R\to FD}$
(c) Gated Fusion

Fig. 2. The overall framework of our proposed RD-GCN model, which contains five parts: 1) Dynamic role-based representation learning module, which extracts high-dimensional dynamic role features from diffusion graphs and encodes the features into low-dimensional role-based node representations; 2) Representation learning from users' interactions module, which learns the representations of user nodes from the static social network and the dynamic diffusion graphs through multiple graph convolutional networks; 3) Gated fusion module, which first integrates the node representations learnt from static and dynamic interactions through a gated mechanism, then further integrates the obtained node representations and role-based node representations to incorporate each individual user's tendency to be influenced by roles through another gated mechanism; 4) Prediction module, which uses a multi-head self-attention module to capture the context-dependency within each cascade and perform diffusion prediction; 5) Joint learning module, which jointly optimizes the loss functions of the dynamic role-based node representations learning component and the information diffusion prediction component.

connected. For example, opinion leaders in different fields may have the same role, but they can be distantly positioned within the network. Hence, directly using the adjacency matrices can not capture the role information. In this paper, to effectively capture the role information embedded in the network structure, we apply a commonly used feature extraction approach ReFex [29]. The features extracted by ReFex are often used as role information [26], [30], [31]. ReFex aggregates node features and the features of the neighborhood nodes in a recursive way to obtain global structure information. Specifically, for each node, ReFex first extracts local features and egonet features based on the node's and egonet's adjacent edges. Local features include the node's in-degree, out-degree and clustering coefficient. And egonet features include the number of edges within the egonet as well as the number of edges entering and leaving the egonet. Then the algorithm aggregates node features and the features of the neighborhood nodes for each node in a recursive way until no additional features can be added. The specific process is illustrated in Algorithm 1. Here, || denotes the concatenation operation, and $N(j)$ denotes the neighborhood of user j. We apply the ReFex approach to the set of diffusion graphs $G_D = \{G_D^1, G_D^2, \ldots, G_D^T\}$ to obtain a set of role features $\{F^1, F^2, \ldots, F^T\}$.

To embed role features into a low-dimensional space, methods such as matrix factorization [30] and auto-encoders [26], [31] can be used. Here, we utilize multiple deep auto-encoders for the role features of different time intervals. The auto-encoders are composed of encoder layers and decoder layers, and here the multi-layer perceptron is adopted for each layer. The encoder layers encode the role features into the low-dimensional role-based node representations, and the decoder layers output the reconstructed features. The distance between input role features and reconstructed features are minimized. As a result, the low-dimensional node representations can preserve the useful information in the role features. For the i-th diffusion graph, given the nodes' role features F^i, K encoder layers are utilized:

$$\begin{aligned} \boldsymbol{h}_{i,j}^1 &= tanh(\boldsymbol{W}^1 \boldsymbol{f}_j^i + \boldsymbol{b}^1) \\ \boldsymbol{h}_{i,j}^k &= tanh(\boldsymbol{W}^k \boldsymbol{h}_{i,j}^{k-1} + \boldsymbol{b}^k) \quad k = 2, 3, \ldots, K \end{aligned} \tag{1}$$

where $\boldsymbol{W}^1, \boldsymbol{b}^1, \boldsymbol{W}^k$ and $\boldsymbol{b}^k$ are learnable parameters, and $\boldsymbol{f}_j^i$ represents the high-dimensional role features of node j in the i-th diffusion graph. Thus, the role-based representations of node j

Algorithm 1: Role Feature Extraction Process.

```
Input: Number of recursion L; Adjacency matrix A.
Output: The role features $F = \{\boldsymbol{f}_j : j = 1, 2, \ldots, |U|\}$.
for $j = 1$ to $|U|$ do
    Calculate the initial features $\boldsymbol{f}_j^0$ of each user node $u_j$, including the node's in-degree, out-degree, clustering coefficient and egonet features;
    $\boldsymbol{f}_j = \boldsymbol{f}_j^0$;
end for
for $l = 1$ to $L$ do
    for $j = 1$ to $|U|$ do
        Aggregate the features of the neighborhood nodes, using sum, mean and concatenation operators as $\boldsymbol{f}_j^l = \sum_{u_k \in N(u_j)} \boldsymbol{f}_k \ || \ \frac{\sum_{u_k \in N(u_j)} \boldsymbol{f}_k}{|N(u_j)|}$;
        $\boldsymbol{f}_j = [\boldsymbol{f}_j \ || \ \boldsymbol{f}_j^l]$;
    end for
end for
return The role features $F = \{\boldsymbol{f}_j : j = 1, 2, \ldots, |U|\}$.
```

in the i-th diffusion graph $\boldsymbol{r}_{i,j}$ can be obtained. Here $\boldsymbol{r}_{i,j} = \boldsymbol{h}_{i,j}^{K}$. And we denote the set of role-based representations of all nodes in the i-th diffusion graph as $R^i = \{\boldsymbol{r}_{i,j} : j = 1, 2, \ldots, |U|\}$. The auto-encoders are used for the diffusion graphs for the T time intervals, and the set of dynamic role-based representations for all user nodes across all time intervals are denoted as $R = \{R^1,\ R^2,\ \ldots,\ R^T\}$.

Then, the reconstructed features $\hat{\boldsymbol{f}}_j^i$ can be obtained by K decoder layers, given as:

$$\begin{aligned} \hat{\boldsymbol{h}}_{i,j}^{k-1} &= tanh(\hat{\boldsymbol{W}}^k \hat{\boldsymbol{h}}_{i,j}^k + \hat{\boldsymbol{b}}^k) \quad k = K, K-1, \ldots, 2 \\ \hat{\boldsymbol{f}}_j^i &= tanh(\hat{\boldsymbol{W}}^1 \hat{\boldsymbol{h}}_{i,j}^1 + \hat{\boldsymbol{b}}^1) \end{aligned} \tag{2}$$

where $\hat{\boldsymbol{W}}^k, \hat{\boldsymbol{b}}^k, \hat{\boldsymbol{W}}^1, \hat{\boldsymbol{b}}^1$ are learnable parameters, $\hat{\boldsymbol{h}}_{i,j}^K = \boldsymbol{r}_{i,j}$.

To maintain the role information of users in the low-dimensional space, the distance between the input role features and reconstructed features need to be minimized. The corresponding loss function is given as follows:

$$\mathcal{L}_{AE} = \sum_{i=1}^{T} \sum_{j=1}^{|U|} \|(\boldsymbol{f}_j^i - \hat{\boldsymbol{f}}_j^i) \odot \boldsymbol{\beta}_j^i\|_2^2 \tag{3}$$

where T represents the number of time intervals, $|U|$ represents the number of user nodes, and $\odot$ denotes the Hadamard product. As the auto-encoders can easily reconstruct the zero elements of the input features, we introduce a correction hyperparameter β to control the contribution of non-zero elements. Specifically, let $f_{j,m}^i$ represents the m-th element of $\boldsymbol{f}_j^i$, and $\beta_{j,m}^i$ represents the m-th element of $\boldsymbol{\beta}_j^i$, if $f_{j,m}^i = 0$, then $\beta_{j,m}^i = 1$; otherwise, $\beta_{j,m}^i = \beta > 1$.

B. Representation Learning From Users' Interactions

1) Static Interactions: Users' static interactions are reflected in the social networks. Researchers [4], [20], [23], [24] have confirmed that introducing social networks is useful for diffusion prediction. Even if some users have not forwarded messages in a cascade, their own preferences can still be inferred by exploring the preferences of their neighbors in the social network. In this paper, a multi-layer graph convolutional network (GCN) is applied in the social friendship network to obtain the representations of user nodes from static interactions of users [65]. In GCNs, node embeddings are produced in a message passing way on the graph, where a node's embedding vector is calculated by recursively aggregating the node embeddings of neighbor nodes. The node embeddings can thus preserve the structural proximity of the graph. A variety of real-world problems such as community detection [44], link prediction [66], information diffusion prediction [18], social recommendation [67], [68] can be solved using the message passing mechanism of GCN, where several graphs like the social networks, diffusion graphs, user-item networks are utilized. Here, the static interactions reflected in the social network are preserved in the embedding space.

Given the social network $G_F = (U, E)$, L layers of graph convolutional operations are applied, and each layer of GCN can be expressed as follows:

$$\boldsymbol{S}_F^{l+1} = \sigma\left(\tilde{\boldsymbol{D}}_F^{-\frac{1}{2}} \tilde{\boldsymbol{A}}_F \tilde{\boldsymbol{D}}_F^{-\frac{1}{2}} \boldsymbol{S}_F^l \boldsymbol{W}_F^l\right) \quad l = 1, 2, \ldots, L \tag{4}$$

where $\tilde{\boldsymbol{A}}_F$ represents the self-looped adjacency matrix of G_F, $\tilde{\boldsymbol{D}}_F$ represents the degree matrix of G_F with self-loops, and $\boldsymbol{W}_F^l \in \mathbb{R}^{d \times d}$ is the trainable weight matrix, where d is the dimension of the node embeddings. The normal distribution is adopted for the random initialization of node embeddings $\boldsymbol{S}_F^0 \in \mathbb{R}^{|U| \times d}$. After learning through GCN, the node representations of static interactions can be obtained from the node embeddings of the L-th layer, denoted as $\boldsymbol{S}_F^L$.

2) Dynamic Interactions: To capture the users' changing interaction preferences, we explore users' dynamic interactions reflected in the users' reposting behaviors. In particular, we obtain the representations of user nodes from the diffusion graphs of different time intervals. Multi-layer graph convolution networks are applied to learn the representations for each diffusion graph. Given the i-th diffusion graph G_D^i, each layer of GCN can be expressed as follows:

$$\boldsymbol{S}_D^{i,l+1} = \sigma\left(\tilde{\boldsymbol{D}_D^i}^{-\frac{1}{2}} \tilde{\boldsymbol{A}}_D^i \tilde{\boldsymbol{D}_D^i}^{-\frac{1}{2}} \boldsymbol{S}_D^{i,l} \boldsymbol{W}_D^{i,l}\right) \quad l = 1, 2, \ldots, L \tag{5}$$

where $\tilde{\boldsymbol{A}}_D^i$ and $\tilde{\boldsymbol{D}_D^i}$ represent the self-looped adjacency matrix and degree matrix of G_D^i, $\boldsymbol{W}_D^{i,l} \in \mathbb{R}^{d \times d}$ is the trainable weight matrix. Similarly, the node embeddings $\boldsymbol{S}_D^{i,0} \in \mathbb{R}^{|U| \times d}$ for the i-th diffusion graph are initialized with a normal distribution, and the node representations for the i-th diffusion graph can be obtained from the node embeddings of the L-th layer, denoted as $\boldsymbol{S}_D^{i,L}$. Finally, we obtain the node representations of dynamic interactions for all time intervals, denoted as $S_D^L = \{\boldsymbol{S}_D^{1,L},\ \boldsymbol{S}_D^{2,L},\ \ldots,\ \boldsymbol{S}_D^{T,L}\}$.

C. Gated Fusion

To capture the importance of different types of interactions, the node representations of static and dynamic interactions are integrated through a gated mechanism. Then, the integrated node representations and role-based node representations are further integrated to learn each user's tendency to be influenced by roles through another gated mechanism.

As the user node representations for the dynamic roles and dynamic interactions are changing, we have designed a strategy to dynamically capture users' roles and preferences: For each cascade, a user has a timestamp recording the time of the user's reposting behavior. Then, the user's role-based node representation and node representation of dynamic interactions during the corresponding time interval are retrieved. For instance, for a given user $u \in U$, the user's role-based node representations for all the time intervals are $\{\boldsymbol{u}_1, \boldsymbol{u}_2,\ \ldots,\ \boldsymbol{u}_T\}$, where T is the number of time intervals. Assuming that user u reposts a message at timestamp t in a cascade, and $t \in [t_2, t_3)$, then $\boldsymbol{u}_2$ is used as the user's role-based node representation in this cascade. Considering the representations learnt from the users' dynamic interactions in each time interval, the node representation of user u for dynamic interactions is extracted in the same way.

We leverage both static dependencies in the social network and the users' dynamic interactions reflected by the diffusion graphs. Instead of directly integrating the corresponding node representations, we introduce a gated fusion mechanism to learn and incorporate the contributions of the two types of interactions, which is described as follows:

$$\begin{aligned}\tilde{\boldsymbol{S}}_D^L &= Lookup(S_D^L)\\ g_{F\to D} &= \sigma(\boldsymbol{W}_g\tilde{\boldsymbol{S}}_D^L + \boldsymbol{U}_g\boldsymbol{S}_F^L)\\ \boldsymbol{S}_{FD} &= g_{F\to D}\odot\tilde{\boldsymbol{S}}_D^L + (1-g_{F\to D})\odot\boldsymbol{S}_F^L \end{aligned}\tag{6}$$

where $Lookup()$ denotes the above designed strategy to capture users' dynamic preferences, $\tilde{\boldsymbol{S}}_D^L$ represents the nodes' representations for dynamic interactions extracted according to the strategy, $g_{F\to D}$ is the gated control matrix, $\sigma()$ denotes the sigmoid activation function, $\boldsymbol{W}_g \in \mathbb{R}^{|U|\times|U|}$ and $\boldsymbol{U}_g \in \mathbb{R}^{|U|\times|U|}$ are the learnable parameters, and $\odot$ represents the Hadamard product. Thus, the integrated user node representations $\boldsymbol{S}_{FD}$ can be obtained.

Considering the fact that the users' tendency to be influenced by roles in the process of information diffusion can be different, we design another gated mechanism to fuse the role-based node representations and the above integrated node representations for user interactions. The gated mechanism is described as follows and the integrated user node representations $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{|U|}]$ can be obtained:

$$\begin{aligned}\tilde{\boldsymbol{R}} &= Lookup(R)\\ m_{h\to FD} &= \sigma(\boldsymbol{W}_m\boldsymbol{S}_{FD} + \boldsymbol{U}_m\tilde{\boldsymbol{R}})\\ \boldsymbol{Z} &= m_{h\to FD}\odot\boldsymbol{S}_{FD} + (1-m_{h\to FD})\odot\tilde{\boldsymbol{R}}\end{aligned}\tag{7}$$

where $Lookup()$ denotes the above designed strategy to capture dynamic roles, $\tilde{\boldsymbol{R}}$ represents users' role-based node representations extracted according to the strategy, $m_{h\to FD}$ is the gated control matrix, $\sigma()$ represents the sigmoid activation function, $\boldsymbol{W}_m \in \mathbb{R}^{|U|\times|U|}$, $\boldsymbol{U}_m \in \mathbb{R}^{|U|\times|U|}$ are learnable parameters, and $\odot$ denotes Hadamard product.

D. Prediction

Until now, the users' static dependencies in the social network and dynamic interactions in the cascades, as well as the dynamic role information are captured and fused to obtain the integrated node representations for each node in the diffusion sequence separately. To further capture context-dependent information within each cascade, we use the masked multi-head self-attention module [69] to process the observed infections in each cascade in parallel.

Given $c^m = \{(u_1^m, t_1^m), (u_2^m, t_2^m), \ldots, (u_{|c^m|}^m, t_{|c^m|}^m)\}$, the m-th cascade, where $|c^m|$ represents the length of the cascade, we locate the corresponding node representations in the integrated node representations $\boldsymbol{Z}$ for each infected user, thus obtaining $\boldsymbol{Z}^m = [\boldsymbol{z}_1^m, \boldsymbol{z}_2^m, \ldots, \boldsymbol{z}_{|c^m|}^m]$. The masked multi-head self-attention module can be formulated as follows and the eventual node representations for all users $\boldsymbol{X} \in \mathbb{R}^{|c^m|\times d}$ can be obtained:

$$\begin{aligned}&Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})\\ &\quad = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}} + \boldsymbol{M}\right)\boldsymbol{V}\\ \boldsymbol{o}_i^m &= Attention(\boldsymbol{Z}^m\boldsymbol{W}_i^Q, \boldsymbol{Z}^m\boldsymbol{W}_i^K, \boldsymbol{Z}^m\boldsymbol{W}_i^V)\\ \boldsymbol{X} &= [\boldsymbol{o}_1^m, \boldsymbol{o}_2^m, \ldots, \boldsymbol{o}_H^m]\boldsymbol{W}^O\end{aligned}\tag{8}$$

where $\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V \in \mathbb{R}^{d\times d_k}$ and $\boldsymbol{W}^O \in \mathbb{R}^{Hd_k\times d}$ are learnable transformation parameters. Here $d_k = d/H$ and H is the number of attention module heads. As later infections can not be used to predict earlier ones, a mask matrix $\boldsymbol{M}$ to prevent leakage of future labels is used, given as:

$$\boldsymbol{M}_{ij} = \begin{cases}0 & i \le j\\ -\infty & otherwise\end{cases}\tag{9}$$

To compute the infection probabilities of all users at each timestamp, a fully-connected neural network with two layers is utilized, given as:

$$\hat{\boldsymbol{y}} = \boldsymbol{W}_2 ReLU(\boldsymbol{W}_1\boldsymbol{X}^T + \boldsymbol{b}_1) + \boldsymbol{b}_2\tag{10}$$

where $\hat{\boldsymbol{y}} \in \mathbb{R}^{|U|\times|c^m|}$, $\boldsymbol{W}_1 \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_2 \in \mathbb{R}^{|U|\times d}$, and $\boldsymbol{b}_1 \in \mathbb{R}^{d\times|c^m|}$, $\boldsymbol{b}_2 \in \mathbb{R}^{|U|\times|c^m|}$ are the learnable parameters.

The cross entropy loss function is adopted as the objective function for the information diffusion prediction component, given as:

$$\mathcal{L}_D = -\sum_{j=2}^{|c^m|}\sum_{i=1}^{|U|}\boldsymbol{y}_{ij}log(\hat{\boldsymbol{y}}_{ij})\tag{11}$$

where $\boldsymbol{y}_{ij} = 1$ indicates that user u_i is the j-th user that reposts a message in the m-th cascade, otherwise $\boldsymbol{y}_{ij} = 0$.

E. Joint Learning

Through the two loss functions of the role-based node representations learning component and information diffusion prediction component, we can jointly optimize the unified deep learning framework. These two components complement and enhance each other, ultimately resulting in a more comprehensive and accurate estimation of the infection probabilities for the user nodes. In addition, we set up a regularizer to prevent the over-fitting problem:

$$\begin{aligned}\mathcal{L}_{reg} = &\sum_{k=1}^{K}(\|\boldsymbol{W}^k\|_2^2 + \|\hat{\boldsymbol{W}}^k\|_2^2 + \|\boldsymbol{b}^k\|_2^2 + \|\hat{\boldsymbol{b}}^k\|_2^2)\\ &+ \sum_{l=1}^{L}\left(\|\boldsymbol{W}_F^l\|_2^2 + \sum_{i=1}^{T}\|\boldsymbol{W}_D^{i,l}\|_2^2\right) + \|\boldsymbol{W}_g\|_2^2 + \|\boldsymbol{U}_g\|_2^2\\ &+ \|\boldsymbol{W}_m\|_2^2 + \|\boldsymbol{U}_m\|_2^2 + \sum_{i=1}^{H}(\|\boldsymbol{W}_i^Q\|_2^2 + \|\boldsymbol{W}_i^K\|_2^2\\ &+ \|\boldsymbol{W}_i^V\|_2^2) + \|\boldsymbol{W}^O\|_2^2 + \|\boldsymbol{W}_1\|_2^2 + \|\boldsymbol{W}_2\|_2^2\\ &+ \|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2\|_2^2\end{aligned}\tag{12}$$

Algorithm 2: Algorithm for Optimizing the RD-GCN.

```
Input: Diffusion graphs for the $T$ time intervals
    $G_D = \{G_D^1, G_D^2, \ldots, G_D^T\}$, social network
    $G_F = (V, E)$, the parameters $\lambda$, $\gamma$, $\beta$.
Output: Infected probabilities $\hat{\boldsymbol{y}}$.
Use Algorithm 1 to generate role features
    $F^i = \{\boldsymbol{f}_j^i : j = 1, 2, \ldots, |U|\}$ for each diffusion graph
    $G_D^i$.
repeat
    Apply multiple auto-encoders to the diffusion graphs
    (1) and (2) to obtain dynamic role-based node
    representations $R$.
    Apply graph convolutional networks (4) and (5) to
    obtain node representations of static interactions $\boldsymbol{S}_F^L$
    and node representations of dynamic interactions
    $\{\boldsymbol{S}_D^{1,L}, \boldsymbol{S}_D^{2,L}, \ldots, \boldsymbol{S}_D^{T,L}\}$.
    Apply gated mechanism (6) and (7) to fuse the above
    three types of node representations and obtain the
    aggregated node representations $\boldsymbol{Z}$.
    Apply multi-head self-attention module (8) to obtain
    the eventual node representations for all users $\boldsymbol{X}$.
    Apply fully-connected neural network (10) to obtain
    users' infected probabilities $\hat{\boldsymbol{y}}$.
    Calculate loss function $\mathcal{L} = \mathcal{L}_{AE} + \lambda\mathcal{L}_D + \gamma\mathcal{L}_{reg}$.
    Use backpropagation to update the parameters.
until converge.
return Infected probabilities $\hat{\boldsymbol{y}}$.
```

Therefore, the final joint loss function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{AE} + \lambda\mathcal{L}_D + \gamma\mathcal{L}_{reg} \tag{13}$$

where λ and γ are the weights of L_D and L_{reg} respectively. The overall process of the entire algorithm is presented in Algorithm 2.

F. Computational Complexity

The algorithm for optimizing the RD-GCN model can be found in Algorithm 2. For using Algorithm 1 to extract high-dimensional role features, the time complexity is $O(T\Delta(|E| + \Delta|U|))$, where $|U|$ is the number of users, $|E|$ is the number of edges, Δ is the number of role features, and T is the number of time intervals. For using auto-encoders to obtain role-based node representations, the time complexity is approximated by $O(T|U|Kd^2)$, where K is the number of layers for the encoders and d represents the dimension of the role-based node representations. For using GCN to extract node representations from the static and dynamic interactions, the time complexity is approximated by $O(TL(|E|d^2 + |U|d))$, where d is the dimension of node representations, which is the same as that for role-based node representations. And L represents the number of GCN layers. For the gated fusion module, the time complexity is $O(T|U|^2\, d)$. For the multi-head self-attention module, the time complexity is $O(d|c^m|(|c^m| + d))$, where $|c^m|$ represents the number of infected users in the cascade. For the fully-connected layer, the time complexity is $O(d|c^m|(d + |U|))$. As $|c^m|$ is smaller than $|U|$, the overall time complexity is $O(T|U|d(Kd + |U| + L) + TL|E|d^2 + T\Delta(|E| + \Delta|U|))$.

TABLE I
STATISTICS OF THE DATASETS

Dataset	Twitter	Douban	MemeTracker
#Users	12627	23123	4709
#Links	309631	348280	-
#Cascades	3442	10602	12661
Avg.Lenth	32.60	27.14	16.24

V. Experiments

In this section, we first briefly describe the data sets, baseline models, evaluation metrics and experimental settings in our experiments. Then, we show the results of our experiments to demonstrate the effectiveness of our proposed model.

A. Datasets

The experiments are conducted on three publicly available datasets used in previous works [4], [23], [24]. The statistics of the datasets are presented in Table I. Here, #Users represents the total number of users, #Links represents the total number of following relationships in the social network, #Cascades represents the total number of cascades, and #Avg.Length represents the average length of the cascades.

1) *Twitter dataset* [70] records the tweets containing URLs and their spreading paths during October 2010. Each URL is regarded as an item of information that is propagated among users. The users' following relationships on Twitter is regarded as social relations.
2) *Douban dataset* [71] collects users' reading status for different books from a Chinese social website. Users can update their own reading status and track the status of others. Each book is regarded as an information item that is propagated among users. If two users engage together in a discussion more than 20 times, they are considered as friends, i.e. a social relationship is constructed.
3) *MemeTracker dataset* [72] collects millions of news stories and blog articles from online websites and tracks the most frequently used quotes and phrases, namely memes, to analyze the migration of memes among different websites. Each website is considered as a node and each meme is considered as an information item. This dataset has no social connections.

B. Baselines

To evaluate the effectiveness of RD-GCN, we compare our proposed model with recent baseline models. The details of the baseline models are as follows:

TopoLSTM: [16] models the spreading paths as dynamic directed acyclic graphs, and extends the basic LSTM model to learn user topology-aware embeddings for prediction.

Deepdiffuse: [17] uses RNN and the attention mechanism to predict the next infected user and time using only the sequence of infected nodes and their infection time.

NDM: [1] alleviates long-term dependence by using convolution neural network (CNN) to extract local influence and encoding global influence with attention mechanism.

SNIDSA: [4] computes the pairwise similarities of different user pairs and adds the structural information into the RNN through a gated mechanism.

FOREST: [23] is a reinforcement learning-based diffusion prediction model, which adds macroscopic diffusion size information into a GRU-based microscopic diffusion model for both macroscopic and microscopic diffusion prediction.

DyHGCN: [24] is an advanced microscopic information diffusion prediction model. It learns dynamic user preferences from discrete diffusion graphs and social relationship from social graph through GCN. On this basis, diffusion prediction is carried out with multiple self-attention modules.

TAN-DRUD: [59] is a non-sequential information cascade model, which enhances information diffusion prediction by capturing the user dependencies of information senders and receivers and incorporating social topology into two-level attention networks.

MS-HGAT: [18] is an information diffusion prediction model that constructs dynamic hypergraphs from the diffusion graph to capture the interaction dependencies of both users and cascades via GCN and self-attention modules.

TABLE II
EXPERIMENTAL RESULTS ON TWITTER DATASET(%)

Models	Twitter					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
DeepDiffuse	4.57	8.80	13.39	3.62	3.79	3.85
TopoLSTM	6.51	15.48	23.68	4.31	4.67	4.79
NDM	21.52	32.23	38.31	14.30	14.80	14.89
SNIDSA	23.37	35.46	43.39	14.84	15.40	15.51
FOREST	26.18	40.95	50.39	17.21	17.88	18.02
DyHGCN	28.48	47.18	58.48	16.78	17.63	17.79
TAN-DRUD	28.13	45.61	55.43	16.35	16.99	17.09
MS-HGAT	28.28	46.69	56.93	17.28	18.13	18.28
RD-GCN	**29.55**	**48.80**	**58.97**	**17.67**	**18.55**	**18.69**

The best results are in boldface.

TABLE III
EXPERIMENTAL RESULTS ON DOUBAN DATASET(%)

Models	Douban					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
Deepdiffuse	9.02	14.93	19.13	4.80	5.07	5.13
TopoLSTM	9.16	14.94	18.93	5.00	5.26	5.32
NDM	10.31	18.87	24.02	5.54	5.93	6.00
SNIDSA	11.81	21.91	28.37	6.36	6.81	6.91
FOREST	14.16	24.79	31.25	7.89	8.38	8.47
DyHGCN	15.69	28.95	36.45	8.42	9.03	9.13
TAN-DRUD	15.27	29.08	37.23	7.79	8.45	8.56
MS-HGAT	16.40	29.45	**37.55**	**8.78**	9.36	9.47
RD-GCN	**16.47**	**29.85**	37.47	8.76	**9.38**	**9.48**

The best results are in boldface.

C. Evaluation Metrics

Following previous studies [4], [23], [24], we view the information diffusion prediction task as an information retrieval task, i.e. we predict the next infected user by ranking the infection probabilities of uninfected users. Thus, two metrics $Hits@k$ and $MAP@k$ are used to evaluate the performance of the models, where k is set as 10, 50 and 100. Higher $Hits$ and MAP values indicate better model performance.

$Hits@k$: The ratio of cases where the true next infected user is within the top k candidate users ranked by the model's predictions, i.e., $Hits@k = \frac{1}{N}\sum_{i=1}^{N} I(rank_i \leq k)$ where N refers to the number of test cases, and $I(\cdot)$ is an indicator function, $rank_i$ represents the ranking position of the true next infected user in the i-th case.

$MAP@k$: is the mean average precision (MAP) on the top k candidate users, i.e., $MAP@k = \frac{1}{N}\sum_{i=1}^{N} AP_{k,i}$, where $AP_{k,i}$ represents the average precision in the i-th test case considering the top k candidate users ranked by the model's predictions. As there is at most one positive case, i.e., the true next infected user, within the top k users in this paper, the value is calculated as $AP_{k,i} = \frac{1}{rank_i} I(rank_i \leq k)$. Compared with $Hits@k$, $MAP@k$ further considers the ranking positions of the next infected users.

D. Experimental Settings

For all the datasets, we randomly choose 80% of the cascades for training, 10% of the cascades for verification, and the remaining 10% of the cascades for testing. For the baselines, we keep the settings in the original papers [1], [4], [16], [17], [18], [23], [24], [59]. For RD-GCN, we choose the optimal parameter configuration based on the performance observed on the validation set and then assess this configuration on the test set. Our model is implemented in Pytorch and the Adam optimizer is utilized to update the model parameters. The parameters in Adam are 0.9 and 0.999 respectively. The learning rate is initialized to 0.001. The dimension of user node representations is set to $d_1 = 64$, the dimension of role-based node representations is set to $d_2 = 64$, and the dimension of hidden layer in GCN is set to $d_3 = 128$. The number of layers of GCN is 2 and the number of layers of auto-encoders is also 2. The hyperparameter β of the role-based node representations learning component is set to 3. The weights of the loss functions are set to $\lambda = 5, \gamma = 0.02$. We divide the original diffusion graph into n time intervals, where $n \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$, with the best setting being 8. The number of heads H in the multi-head attention module is selected from $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$, with the best setting being 14.

E. Experimental Results

We evaluate the effectiveness of RD-GCN for the information diffusion prediction task on three public datasets. The overall performance of RD-GCN and the baseline models is shown in Tables II, III and IV. We exclude TopoLSTM, SNIDSA and MS-HGAT from Table IV due to the absence of an underlying

TABLE IV
EXPERIMENTAL RESULTS ON MEMETRACKER DATASET(%)

Models	MemeTracker					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
DeepDiffuse	13.93	26.50	34.77	8.14	8.69	8.80
NDM	25.44	42.19	51.14	13.57	14.33	14.46
FOREST	29.43	47.41	56.77	16.37	17.21	17.34
DyHGCN	29.63	48.78	58.78	16.33	17.21	17.36
TAN-DRUD	29.13	**49.50**	59.38	15.14	16.06	16.20
RD-GCN	**29.93**	49.27	**59.51**	**16.39**	**17.28**	**17.43**

The best results are in boldface.

social network in the MemeTracker dataset. It is evident that RD-GCN outperforms the baselines on all datasets and across all evaluation metrics, confirming the validity of the proposed model.

Specifically, we obtain the following observations:

- The performance of TopoLSTM, DeepDiffuse, and NDM on these datasets are not satisfactory. These models primarily focus on the dependencies within the current cascades, and neglect the influence of users' social network. In reality, users' social relationships can impact their preferences, as they tend to repost the messages shared by their friends. In comparison, SNIDSA, FOREST, DyHGCN, TAN-DRUD, MS-HGAT and our model RD-GCN all take into account the social network for information diffusion prediction, achieving superior performance. This indicates that it is essential to consider the static interactions in the social network for diffusion prediction.
- Compared to SNDISA and FOREST, the models of DyHGCN, TAN-DRUD, MS-HGAT and our proposed RD-GCN exhibit significant improvement of nearly 3% to 15% across all evaluation metrics. We find that while both SNIDSA and FOREST take into account the social relationships among users, they model diffusion paths as sequences and rely on RNN-based methods for diffusion prediction. As a result, these approaches may not effectively capture the dynamic dependencies among users. On the contrary, DyHGCN, MS-HGAT, and our RD-GCN all use graphs to capture the dynamic interactions of users and utilize graph neural networks to jointly encode the diffusion graphs. And TAN-DRUD proposes a two-level attention network to depict the dynamic interactions in the diffusion graph. Consequently, these models can enhance prediction performance. The experiments demonstrate that representing diffusion paths as graphs is a wise choice to capture the dynamic dependencies among users.
- As our model is built upon DyHGCN, we compare the performance with DyHGCN and the results show that RD-GCN outperforms DyHGCN on all three datasets, achieving the performance improvement ranging from approximately 0.83% to 5.30%. This demonstrates that considering the role information of users in the diffusion process is beneficial for diffusion prediction. Also, RD-GCN exhibits the most significant improvement in $Hits@10$ and $MAP@10$. This can be attributed to the understanding that users with higher role status are more likely to influence others, resulting in more accurate predictions for the activations of top-ranked users.
- While our model builds upon DyHGCN with improvement, it also outperforms the two latest models, TAN-DRUD and MS-HGAT. TAN-DRUD, MS-HGAT and our proposed RD-GCN all incorporate the diffusion graphs and social network, but they consider different additional aspects. MS-HGAT builds upon DyHGCN to characterize the interaction dependencies of both users and cascades, TAN-DRUD captures the user dependencies of information senders and receivers, and our model RD-GCN captures the role information in the cascades. Comparison with them can reflect the relative importance of different aspects. It is evident that our model also consistently outperforms the two latest models on each dataset, especially the $Hits@k$ metrics on the Twitter dataset. Compared to MS-HGAT, our model RD-GCN demonstrates a notable performance boost on the Twitter dataset, with a nearly 4.2% increase on the $Hits@k$ metrics and a 2.3% increase on the $MAP@k$ metrics. Also, the performance of TAN-DRUD and MS-HGAT are not stable, indicating the aspects they consider are not suitable for all situations. For example, TAN-DRUD and MS-HGAT exhibit apparently weaker performance than DyHGCN for the $Hits@k$ metrics on the Twitter dataset. In addition, TAN-DRUD's performance on $MAP@k$ metrics across all the three datasets are unsatisfactory, indicating that the influenced users appear in the top-k list, but are ranked relatively lower for TAN-DRUD. This is possibly due to its inability to directly use the existing diffusion graphs. The above results indicate that our model considering role information exhibits more pronounced performance in predicting the next infected user. Our experiments demonstrate that the roles learned by the RD-GCN model makes a dominant contribution to information prediction.

F. Ablation Study

To investigate the contributions of different components, we conduct a series of ablation studies on four variants of RD-GCN using Twitter dataset, Douban dataset and MemeTracker dataset. The variants are defined as follows:

w/o Role: To assess the impact of role information on diffusion prediction, we remove the dynamic role-based representation learning module.

w/o DR: To analyze the contribution of modeling the dynamic nature of role information for diffusion prediction, we extract role features directly from the original diffusion graph without considering dynamic information.

w/o GF: To investigate whether the gated fusion mechanism has captured distinct trends of how different people are influenced by roles, we replace the gated fusion mechanism with a simple summation mechanism to aggregate the role-based node representations and the node representations for user interactions in this variant.

TABLE V
ABLATION STUDY ON TWITTER DATASET(%)

Models	Twitter					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
RD-GCN	**29.55**	**48.80**	**58.97**	**17.67**	**18.55**	**18.69**
w/o Role	28.50	46.34	56.47	16.48	17.28	17.44
w/o DR	28.31	45.17	55.15	17.38	18.15	18.29
w/o GF	28.48	47.25	58.18	17.12	17.97	18.13
w/o JL	28.57	47.32	58.28	16.90	17.75	17.90

The best results are in boldface.

TABLE VI
ABLATION STUDY ON DOUBAN DATASET(%)

Models	Douban					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
RD-GCN	**16.47**	**29.85**	**37.47**	**8.76**	**9.38**	**9.48**
w/o Role	16.40	29.51	37.11	8.69	9.23	9.40
w/o DR	16.29	29.07	36.51	8.74	9.35	9.46
w/o GF	16.36	29.54	37.16	8.73	9.31	9.43
w/o JL	16.20	29.19	36.98	8.63	9.22	9.33

The best results are in boldface.

TABLE VII
ABLATION STUDY ON MEMETRACKER DATASET(%)

Models	MemeTracker					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
RD-GCN	**29.93**	**49.27**	**59.51**	**16.39**	**17.28**	**17.43**
w/o Role	29.58	48.43	58.41	16.13	16.99	17.13
w/o DR	28.68	47.99	58.06	15.86	16.76	16.90
w/o GF	29.05	48.24	58.40	16.19	17.09	17.23
w/o JL	29.75	49.15	59.35	16.29	17.20	17.34

The best results are in boldface.

w/o JL: To investigate whether the role-based node representations learning component and information diffusion prediction component can mutually reinforce each other, we first train the role-based node representations only on the diffusion graphs and then directly apply the obtained role-based node representations for the diffusion prediction task.

Tables V, VI and VII present the performance of RD-GCN and its different variants. The results demonstrate that RD-GCN outperforms all other variants, affirming the validity of our model design. Specifically, we have obtained the following observations:

- The model exhibits a noticeable drop in performance when the dynamic role-based representation learning module is removed. This suggests that role information indeed enhances diffusion prediction, thereby affirming the validity of our proposed approach. Compared with the other two datasets, in the Douban dataset, the impact of removing role information seems less significant. We observe that in the Douban dataset, individuals are influenced by local book friends in many situations, and the influence of individuals with high role status is not as pronounced. For instance, when a person's majority of book friends prefer a particular book, although famous reviewers do not recommend that book, this person is inclined to read the book.
- When the dynamic role information is replaced with global static role information, the values have dropped apparently. This is due to the fact that the use of the entire diffusion graph incorporates the overall role information, rather than the role information specific to the users' activated time periods. This outcome underscores the significance of considering dynamic role information, as each individual's role information varies over time.
- The values for all the metrics have dropped by about 3% after removing the gated fusion module. This indicates that the gated fusion mechanism we have crafted does improve the effectiveness of diffusion prediction. It is able to determine which people are more likely to be influenced by the roles and which people are less likely to be influenced by the roles.
- The role-based node representations learning component and information diffusion prediction component can indeed manage to enhance each other and improve the performance of diffusion prediction.

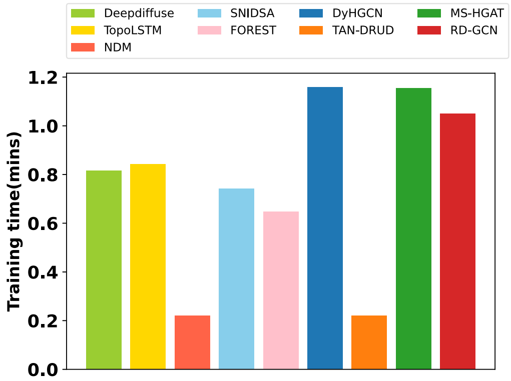
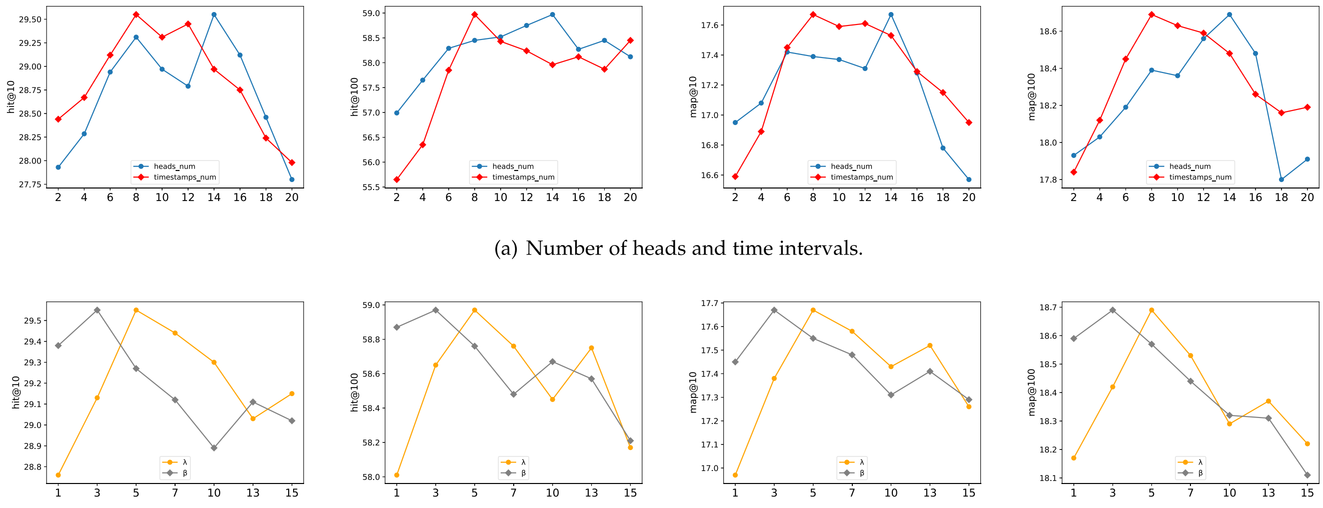


Fig. 3. Average training time per epoch for RD-GCN and the baseline models on the Twitter dataset. The models are sorted from left to right based on their performance.

G. Training Time Analysis

In this section, we investigate the time efficiency issue and compare the run-time of RD-GCN and the baseline models. Fig. 3 shows the average training time per epoch for different models on the Twitter dataset, with the models sorted from left to right based on their performance.

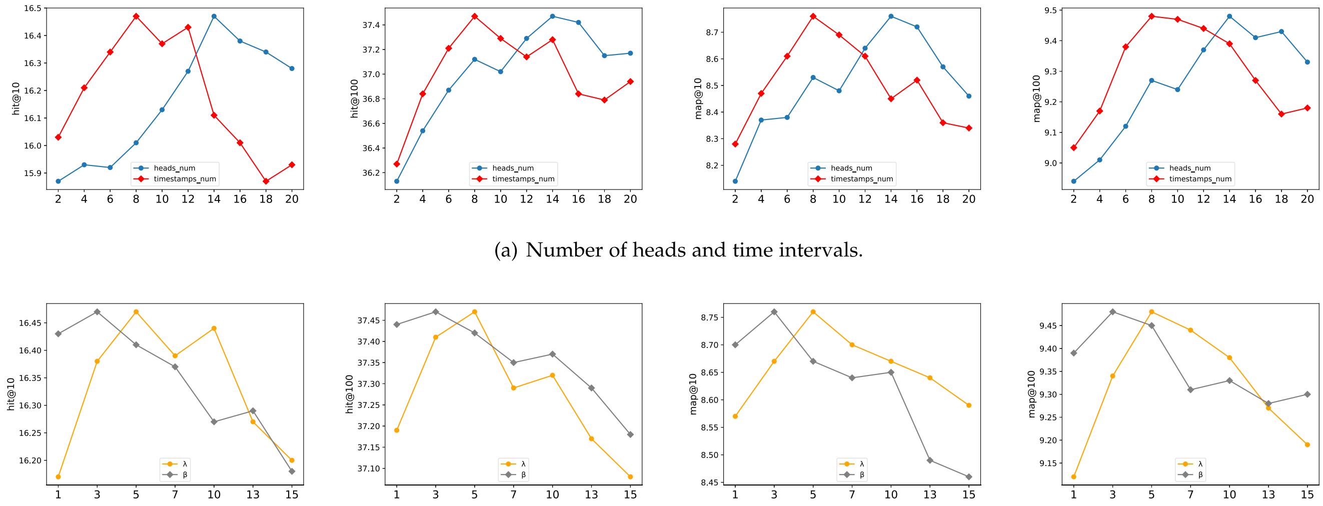
Among these models, NDM and TAN-DRUD have the least average training time per epoch, with just 0.221 and 0.222 minutes respectively. However, their performance is not particularly good. NDM employs the convolutional networks module, which is more efficient when processing large graph datasets. TAN-DRUD utilizes the Node2vec algorithm to learn node embeddings from social networks, and the embedding generation time is relatively less compared to that of the graph neural network counterpart. In comparison, Deepdiffuse, TopoLSTM, SNIDSA, and FOREST have relatively longer training time, with 0.817 minutes, 0.843 minutes, 0.742 minutes and 0.648 minutes respectively. The training time of these models is quite



(a) Number of heads and time intervals.

(b) Values of λ and β.

Fig. 4. Parameter analysis on Twitter dataset.



(a) Number of heads and time intervals.

(b) Values of λ and β.

Fig. 5. Parameter analysis on Douban dataset.

similar because these baselines all rely on LSTM-based models or their variants to process the diffusion sequences. DyHGCN, MS-HGAT and RD-GCN, on the other hand, have the longest average training time, with 1.158 minutes, 1.155 minutes and 1.05 minutes respectively. These three models all consider dynamic diffusion graphs and social graphs, although they use a multi-head self-attention mechanism to process the sequences instead of using LSTM, their run-time is longer compared to other baselines. Nevertheless, the increase is not much, only about 35% compared to the LSTM-based models. This shows that the increase in training time is reasonable when considering dynamic diffusion graphs and social graphs. Although our model RD-GCN uses the joint training module, the actual training time of RD-GCN is similar to that of DyHGCN and MS-HGAT.

H. Parameter Analysis

In this section, we conduct sensitivity analysis of the hyperparameters using the Twitter and Douban datasets. The impact of different values for hyperparameters on model performance are assessed. Specifically, the values for hyperparameters β in the role-based node representations learning component, λ in the loss function, the number of discrete time intervals T, and the number of heads H in the multi-head attention module are adjusted. The observed results are depicted in Figs. 4 and 5.

In Figs. 4(a) and 5(a), we assess the impact of hyperparameters β and λ, and have the following observations:

- The hyperparameter λ in the loss function controls the relative importance of the role-based node representations learning component and information diffusion prediction component in the objective function of optimization. As the value of λ increases, the model's performance initially increases and then decreases. This suggests that as λ increases, the role of dynamic role features in enhancing the performance of diffusion prediction task becomes important. As a consequence, the performance increases at the first stage. However, when $\lambda > 5$, the model becomes overly reliant on the prediction performance, and the learning of role-based node representations which preserves the high-dimensional role features is neglected, leading to a decreased generalization ability.

- The hyperparameter β is used to control the contribution of nonzero elements in the reconstruction error in the the role-based node representations learning component. As depicted in the figure, the model's performance is optimized when $\beta = 3$.

In Figs. 4(b) and 5(b), we compare the prediction performance across different numbers of heads and time intervals, and we have made the following observations:

- The entire historical information diffusion time period is divided into T time intervals, and the original diffusion graph is divided into T subgraphs according to the time intervals to preserve the dynamic interactions. The results show that the number of time intervals can not be too small or large. The best performance of the model is achieved when the number is 8. If the number is too small and each time interval is very large, the subgraphs cannot precisely reflect the users' dynamic information at each stage. If the number is too large, there will be only a few user interactions in each time interval.
- As the number of heads H in the multi-head self-attention module increases, the performance keeps increasing. The best performance is achieved at $H = 14$. It indicates that our model does capture richer information with more heads when $H \leq 14$. However, as the number of heads exceeds 14, the performance drops due to overfitting.

I. *Case Study*

In this section, we conduct a case study on the MemeTracker network to intuitively demonstrate that the role-based node representations we have obtained can reflect the role information of nodes. We employ the K-means clustering algorithm to group the nodes based on the role-based node representations and assign role labels according to the clusters. Thus, the effectiveness of the learnt role-based node representations can be verified through the clustering results. The observed results for the MemeTracker network are depicted in Fig. 6, where the nodes represent the online websites and the edges represent the reference interactions. The nodes with the same color have the same role label. Due to the large scale of the MemeTracker dataset, we sample several important websites and some of their 2-hop neighbors in the network to visualize the role information. It is worth noting that the three pink nodes in the figure, i.e., *The New York Times*, *The Times of India*, and *The Bloomberg News*, occupy the same position, as they all represent some of the most famous news websites. However, within the entire MemeTracker dataset, these three nodes are distantly separated from each other and lack any direct connections. Furthermore, some weblog sites or small news websites have also been grouped together, represented by the green nodes in the figure. They are usually connected to the pink nodes. Also, the grey nodes are typically smaller news and weblog sites, and they are mostly connected to the green nodes. This demonstrates that the learnt role-based node representations can indeed help group the nodes based on their role-based similarities, where the most famous news websites have the same role label although they are typically not closely connected.

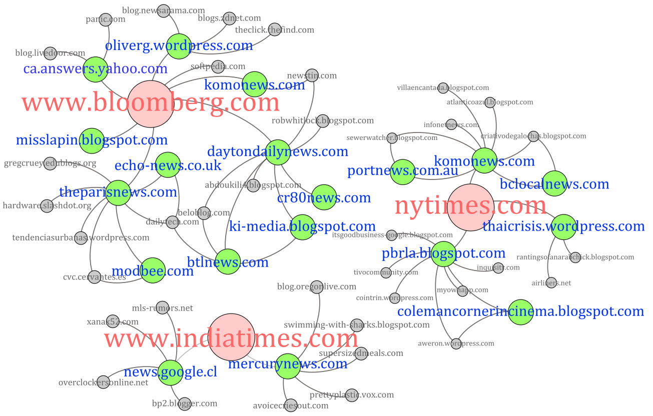


Fig. 6. Role discovery on the MemeTracker dataset, where the nodes with the same color have the same role label.

VI. Conclusion

In this paper, we incorporate structural similarity of the networks, namely user roles, and propose a novel Role-based Dynamic Graph Convolutional Network (RD-GCN) for information diffusion prediction. RD-GCN captures the dependencies of dynamic user roles and information diffusion, and jointly learns the two parts. In particular, to dynamically capture users' preferences and roles, the original diffusion graph is divided into subgraphs according to the timestamps. Then, the users' preferences and roles are captured based on those sequential subgraphs and social networks. Also, gated mechanisms are designed to incorporate users' different tendencies to be influenced by roles. The experimental results show that our proposed method outperforms the state-of-the-art diffusion prediction methods on three public datasets.

There are still some interesting phenomena to be explored for future work. For instance, other structural characteristics can be considered, such as the rich-club effect, which refers to the impact of closely connected celebrities serving as a network's backbone. Alternatively, whether the user roles also depend on different types of diffusion contents is another worthwhile research direction. In addition, the social network remains static in this study, and the dynamic changes in social networks can also be incorporated.

REFERENCES

[1] C. Yang, M. Sun, H. Liu, S. Han, Z. Liu, and H. Luan, "Neural diffusion model for microscopic cascade study," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1128–1139, Mar. 2021.

[2] Q. Cao, H. Shen, K. Cen, W. Ouyang, and X. Cheng, "DeepHawkes: Bridging the gap between prediction and understanding of information cascades," in *Proc. 26th ACM Int. Conf. Inf. Knowl. Manage.*, 2017, pp. 1149–1158.

[3] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, "Cascade dynamics modeling with attention-based recurrent neural network," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2985–2991.

[4] Z. Wang, C. Chen, and W. Li, "A sequential neural information diffusion model with structure attention," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1795–1798.

[5] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 2077–2085.

[6] C. Yuan, W. Zhou, Q. Ma, S. Lv, J. Han, and S. Hu, "Learning review representations from user and product level information for spam detection," in *Proc. IEEE Int. Conf. Data Mining*, 2019, pp. 1444–1449.

[7] Y. Chen, J. Sui, L. Hu, and W. Gong, "Attention-residual network with CNN for rumor detection," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 1121–1130.

[8] H. Guo, J. Cao, Y. Zhang, J. Guo, and J. Li, "Rumor detection with hierarchical social attention network," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 943–951.

[9] T. Bian et al., "Rumor detection on social media with bi-directional graph convolutional networks," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 549–556.

[10] Y. Cheng, J. Liu, and X. Yu, "Online social trust reinforced personalized recommendation," *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 457–467, 2016.

[11] Y. Liu, S. Yang, Y. Zhang, C. Miao, Z. Nie, and J. Zhang, "Learning hierarchical review graph representations for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 658–671, Jan. 2023.

[12] K. Saito, M. Kimura, K. Ohara, and H. Motoda, "Learning continuous-time information diffusion model for social behavioral data analysis," in *Proc. 1st Asian Conf. Mach. Learn.*, 2009, pp. 322–337.

[13] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 4, pp. 1–37, 2012.

[14] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in *Proc. 12th Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst.*, 2008, pp. 67–75.

[15] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.

[16] J. Wang, V. W. Zheng, Z. Liu, and K.-C. Chang, "Topological recurrent neural network for diffusion prediction," in *Proc. 17th Int. Conf. Data Mining*, 2017, pp. 475–484.

[17] M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrishnan, "DeepDiffuse: Predicting the 'who' and 'when' in cascades," in *Proc. 18th Int. Conf. Data Mining*, 2018, pp. 1055–1060.

[18] L. Sun, R. Yuan, Z. Xiangbo, Y. Lan, and S. Yu, "MS-HGAT: Memory-enhanced sequential hypergraph attention network for information diffusion prediction," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 4156–4164.

[19] D. Wang et al., "Cascade-enhanced graph convolutional network for information diffusion prediction," in *Proc. 27th Int. Conf. Database Syst. Adv. Appl.*, 2022, pp. 615–631.

[20] Z. Wang and W. Li, "Hierarchical diffusion attention network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3828–3834.

[21] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1555–1564.

[22] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "DeepInf: Social influence prediction with deep learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2110–2119.

[23] C. Yang, J. Tang, M. Sun, G. Cui, and Z. Liu, "Multi-scale information diffusion prediction with reinforced recurrent networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4033–4039.

[24] C. Yuan, J. Li, W. Zhou, Y. Lu, X. Zhang, and S. Hu, "DyHGCN: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2021, pp. 347–363.

[25] N. Huang, G. Zhou, M. Zhang, and M. Zhang, "MC-RGCN: A multichannel recurrent graph convolutional network to learn high-order social relations for diffusion prediction," in *Proc. IEEE Int. Conf. Data Mining*, 2021, pp. 1108–1113.

[26] P. Jiao, Q. Tian, W. Zhang, X. Guo, D. Jin, and H. Wu, "Role discovery-guided network embedding based on autoencoder and attention mechanism," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 365–378, Jan. 2023.

[27] T. Lou and J. Tang, "Mining structural hole spanners through information diffusion in social networks," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 825–836.

[28] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, "Who says what to whom on twitter," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 705–714.

[29] K. Henderson et al., "It's who you know: Graph mining using recursive structural features," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2011, pp. 663–671.

[30] K. Henderson et al., "RolX: Structural role extraction and mining in large graphs," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 1231–1239.

[31] H. Ke et al., "Deep mutual encode model for network embedding from structural identity," *IEEE Access*, vol. 7, pp. 177484–177496, 2019.

[32] P. Jiao, X. Guo, T. Pan, W. Zhang, Y. Pei, and L. Pan, "A survey on role-oriented network embedding," *IEEE Trans. Big Data*, vol. 8, no. 4, pp. 933–952, Aug. 2022.

[33] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Commun. ACM*, vol. 53, no. 8, pp. 80–88, 2010.

[34] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of youtube videos," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 365–374.

[35] L. Weng, F. Menczer, and Y.-Y. Ahn, "Virality prediction and community structure in social networks," *Sci. Rep.*, vol. 3, no. 1, pp. 1–6, 2013.

[36] H. Shen, D. Wang, C. Song, and A.-L. Barabási, "Modeling and predicting popularity dynamics via reinforced poisson processes," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 291–297.

[37] J. Gao, H. Shen, S. Liu, and X. Cheng, "Modeling and predicting retweeting dynamics via a mixture process," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 33–34.

[38] C. Li, J. Ma, X. Guo, and Q. Mei, "DeepCas: An end-to-end predictor of information cascades," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 577–586.

[39] C. Gou, H. Shen, P. Du, D. Wu, Y. Liu, and X. Cheng, "Learning sequential features for cascade outbreak prediction," *Knowl. Inf. Syst.*, vol. 57, no. 3, pp. 721–739, 2018.

[40] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, and F. Zhang, "Information diffusion prediction via recurrent cascades convolution," in *Proc. 35th Int. Conf. Data Eng.*, 2019, pp. 770–781.

[41] Q. Cao, H. Shen, J. Gao, B. Wei, and X. Cheng, "Popularity prediction on social platforms with coupled graph neural networks," in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 70–78.

[42] X. Jia, J. Shang, D. Liu, H. Zhang, and W. Ni, "HeDAN: Heterogeneous diffusion attention network for popularity prediction of online content," *Knowl.-Based Syst.*, vol. 254, 2022, Art. no. 109659.

[43] X. Chen, F. Zhang, F. Zhou, and M. Bonsangue, "Multi-scale graph capsule with influence attention for information cascades prediction," *Int. J. Intell. Syst.*, vol. 37, no. 3, pp. 2584–2611, 2022.

[44] S. Yuan, H. Zeng, Z. Zuo, and C. Wang, "Overlapping community detection on complex networks with graph convolutional networks," *Comput. Commun.*, vol. 199, pp. 62–71, 2023.

[45] C. Gao, Z. Yin, Z. Wang, X. Li, and X. Li, "Multilayer network community detection: A novel multi-objective evolutionary algorithm based on consensus prior information [feature]," *IEEE Comput. Intell. Mag.*, vol. 18, no. 2, pp. 46–59, May 2023.

[46] L. Yu, X. Xu, G. Trajcevski, and F. Zhou, "Transformer-enhanced hawkes process with decoupling training for information cascade prediction," *Knowl.-Based Syst.*, vol. 255, 2022, Art. no. 109740.

[47] Y. Wang, X. Wang, Y. Ran, R. Michalski, and T. Jia, "CasSeqGCN: Combining network structure and temporal sequence to predict information cascades," *Expert Syst. Appl.*, vol. 206, 2022, Art. no. 117693.

[48] L. Wu, H. Wang, E. Chen, Z. Li, H. Zhao, and J. Ma, "Preference enhanced social influence modeling for network-aware cascade prediction," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 2704–2708.

[49] Y. Zeng and K. Xiang, “Persistence augmented graph convolution network for information popularity prediction,” *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 6, pp. 3331–3342, Nov./Dec. 2023.

[50] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, “Uncovering the temporal dynamics of diffusion networks,” in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 561–568.

[51] Z. Zhang, W. Zhao, J. Yang, C. Paris, and S. Nepal, “Learning influence probabilities and modelling influence diffusion in twitter,” in *Proc. 28th Int. Conf. World Wide Web*, 2019, pp. 1087–1094.

[52] W. Lee, J. Kim, and H. Yu, “CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing,” in *Proc. 12th Int. Conf. Data Mining*, 2012, pp. 960–965.

[53] S. Bourigault, S. Lamprier, and P. Gallinari, “Representation learning for information diffusion through social networks: An embedded cascade model,” in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, 2016, pp. 573–582.

[54] Y. Zhang, T. Lyu, and Y. Zhang, “COSINE: Community-preserving social network embedding from information diffusion cascades,” in *Proc. 32th AAAI Conf. Artif. Intell.*, 2018, pp. 2620–2627.

[55] S. Xiao, J. Yan, X. Yang, H. Zha, and S. M. Chu, “Modeling the intensity function of point process via recurrent neural networkss,” in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1597–1603.

[56] A. Sankar, X. Zhang, A. Krishnan, and J. Han, “Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction,” in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 510–518.

[57] S. Feng et al., “H-Diffu: Hyperbolic representations for information diffusion prediction,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 8784–8798, Sep. 2023.

[58] S. Xu, L. Zhou, J. Xu, L. Wang, and H. Chen, “MSIDP: Multi-scale information diffusion prediction with timestamp information and wide dispersion,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2022, pp. 1–10.

[59] B. Liu, D. Yang, Y. Shi, and Y. Wang, “Improving information cascade modeling by social topology and dual role user dependency,” in *Proc. 27th Int. Conf. Database Syst. Adv. Appl.*, 2022, pp. 425–440.

[60] L. Pan, Y. Xiong, B. Li, T. Huang, and W. Wan, “Feature attenuation reinforced recurrent neural network for diffusion prediction,” *Appl. Intell.*, vol. 53, no. 2, pp. 1855–1869, 2023.

[61] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, “Struc2vec: Learning node representations from structural identity,” in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 385–394.

[62] N. K. Ahmed et al., “Role-based graph embeddings,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2401–2415, May 2022.

[63] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, “Deep recursive network embedding with regular equivalence,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2357–2366.

[64] T. Pan, W. Wang, M. Shao, Y. Sun, and P. Jiao, “Role-oriented dynamic network embedding,” in *Proc. IEEE Int. Conf. Big Data*, 2022, pp. 1070–1079.

[65] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. 5th Int. Conf. Learn. Representations*, 2017, pp. 1–14.

[66] C. Gao, J. Zhu, F. Zhang, Z. Wang, and X. Li, “A novel representation learning for dynamic graphs based on graph convolutional networks,” *IEEE Trans. Cybern.*, vol. 53, no. 6, pp. 3599–3612, Jun. 2023.

[67] L. Wu et al., “A neural influence diffusion model for social recommendation,” in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 235–244.

[68] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, “DiffNet++ : A neural influence and interest diffusion network for social recommendation,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4753–4766, Oct. 2022.

[69] A. Vaswani et al., “Attention is all you need,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[70] N. O. Hodas and K. Lerman, “The simple rules of social contagion,” *Sci. Rep.*, vol. 4, no. 1, 2014, Art. no. 4343.

[71] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li, “Comsoc: Adaptive transfer of user behaviors over composite social network,” in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 696–704.

[72] J. Leskovec, L. Backstrom, and J. Kleinberg, “Meme-tracking and the dynamics of the news cycle,” in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 497–506.

Jingkai Ye was born in Taizhou, China, in 1999. He received the B.Eng. degree in network engineering from Wenzhou University, Wenzhou, China, in 2017. He is currently working toward the M.S. degree with the School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, China. His research interests include graph neural networks and information diffusion prediction.

Qing Bao received the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong, in 2016. She is currently an Associate Professor with the School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, China. Her research interests include social network analysis, graph data mining, and health informatics. Dr. Bao was the recipient of the Best Student Paper Award at the 2013 IEEE/WIC/ACM International Conference on Web Intelligence.

Ming Xu received the M.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2000 and 2004, respectively. He is currently a Full Professor with Hangzhou Dianzi University, Hangzhou. His research interests include digital forensics and network security.

Jian Xu received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2004. He is currently a Full Professor with Hangzhou Dianzi University, Hangzhou. His research interests include machine learning, data mining, and social network analysis.

Hongjun Qiu received the B.Sc. degree in computer science and technology from Beijing Forestry University, Beijing, China in 2003, and the Ph.D degree in computer application from the Beijing University of Technology, Beijing, in 2010. She is currently a Lecturer with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China. Her research interests include complex systems/networks, health informatics, and autonomy-oriented computing.

Pengfei Jiao received the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2018. From 2018 to 2021, he was a Lecturer with the Center of Biosafety Research and Strategy of Tianjin University. He is currently a Professor with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China. His research interests include complex network analysis and its applications.