SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2026, Vol. 69, Iss. 9, 192101:1–192101:16
https://doi.org/10.1007/s11432-025-4727-6

Multi-scale diffusion prediction based on user hierarchical latent interactions and cross-task feature similarity

Tianyang SHAO[1], Xiang ZHAO[2*], Jichao LI[3] & Xin LU[3]

[1]*National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha 410073, China*
[2]*Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China*
[3]*College of Systems Engineering, National University of Defense Technology, Changsha 410073, China*

Received 11 March 2025/Revised 8 September 2025/Accepted 17 December 2025/Published online 31 August 2026

Abstract Information diffusion prediction comprises macroscopic and microscopic prediction tasks. The former aims to estimate the overall impact of the information propagation process, while the latter focuses on predicting the next participating user, both of which are crucial to understanding how information spreads in a network. Hence, as a recent trend, it is important to investigate multi-scale prediction models. Nevertheless, conventional methods primarily focus on extracting shared features but fail to consider the complex interactions between users and cascades, as well as the feature similarity of the same user across tasks. In this connection, we propose a multi-dimensional feature interaction (MDFI) prediction model with two novel designs. Specifically, to model the latent hierarchical interactions between users as well as users and cascades, we construct a heterogeneous weighted user-cascade graph at the macro-level. At the micro-level, we account for the social homophily of users in the social network and develop the cascade neighbor contrastive learning as a constraint under the multi-task learning framework. Experimental results on four publicly available datasets demonstrate that MDFI exhibits superior performance to all competing models.

Keywords social network, information diffusion, heterogeneous graph, contrastive learning, graph neural network

Citation Shao T Y, Zhao X, Li J C, et al. Multi-scale diffusion prediction based on user hierarchical latent interactions and cross-task feature similarity. Sci China Inf Sci, 2026, 69(9): 192101, https://doi.org/10.1007/s11432-025-4727-6

1 Introduction

The rapid development of online social platforms, such as Facebook, Twitter, and Weibo, has significantly broadened access to information for individuals and facilitated the rapid propagation of information, which gives rise to the phenomena of information cascades. Modeling information cascades is crucial to understanding the mechanism of information diffusion, which in turn is of paramount importance for various downstream tasks such as fake news detection [1], viral marketing [2], and hotspots detection [3].

As shown in Figure 1, there are two distinct types of prediction tasks associated with information cascades: (1) macroscopic prediction [4, 5], which aims to estimate the final size of the cascade and predict outbursts; and (2) microscopic prediction [6, 7], which aims to predict the next user to participate in the cascade forwarding. In contrast, the former emphasizes cascade modeling, while the latter focuses on individual users, and hence, methods for macroscopic prediction may not be suitable for user prediction at the micro-level. Nevertheless, both tasks are of importance to understanding how information spreads in a network.

Recently, there has been pioneering work, represented by Forest [8], DMT-LIC [9], and MINDS [10], trying to perform multi-scale diffusion prediction (i.e., both macroscopic and microscopic), by enabling microscopic prediction models to also perform macroscopic prediction [11]. The proposed solutions are viable, however, they do not adequately recognize the synergistic enhancements that exist between the two related tasks. Subsequently, DMT-LIC [9] and MINDS [10] were proposed, which employ a shared representation layer to concurrently model the social network and information diffusion processes. In particular, MINDS builds upon DMT-LIC by devising an adversarial training module to resolve contamination and redundancy between shared features and task-specific ones.

It is noted that prior work concentrates on learning strategies for multi-scale prediction models, and the existing modeling of diffusion cascades is inadequately considered and less discussed. First, existing methods primarily

* Corresponding author (email: xiangzhao@nudt.edu.cn)

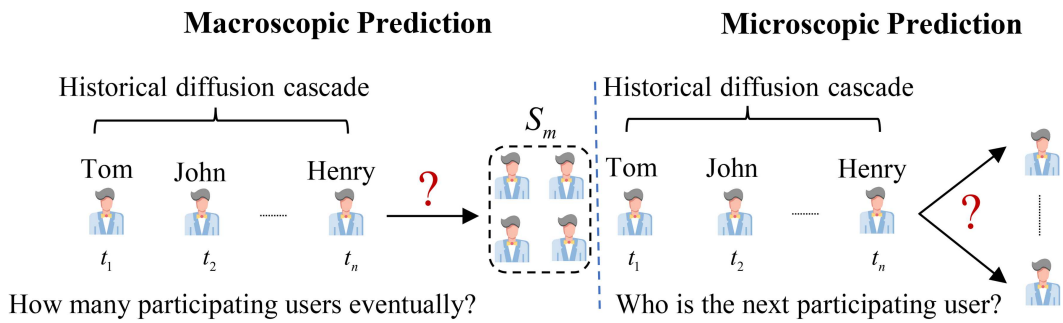

Figure 1 (Color online) Illustration of macroscopic prediction on cascade size (left) and microscopic prediction on next participating users (right).

take in account explicit interactions (i.e., between user-user). The coarse modeling simply neglects the latent interactions between users and cascades, which can still be common on real-world social platforms. For example, a user may retweet a post from a stranger, thus joining in some information cascades, and if the user later becomes interested in his/her previous posts, one may further retweet those historical posts. In this way, the user indirectly engages with other cascades that are initially unrelated. The indirect influence of user reposting behaviors and the nonlinear nature of information diffusion require necessary care handling during cascade modeling. Second, existing methods learn a multi-scale prediction model by incorporating a multi-task learning framework, where some shared features beneficial to macroscopic and microscopic predictions are extracted. Intuitively, the features of a user across different tasks bear a certain degree of similarity, and this can be exploited to highlight the inherent consistency in user behavior across different scales—a consistency that has largely been overlooked until now. Under the multi-task learning framework, this can be leveraged to serve as additional micro-level connections between the two prediction tasks.

Information cascades represent user-driven propagation sequences, where participants are influenced not only by their immediate cascades but also by latent dependencies across cascades, which exhibits structural parallels with semantic and syntactic information capturing in natural language processing (NLP) [12]: just as words derive meaning from both intra-sentence syntactic relations and inter-sentence semantic dependencies across a corpus, social media users exhibit behavioral patterns shaped by both local cascade dynamics and global information diffusion contexts. Motivated by this analogy, we conceive the notion of user-cascade graph as a heterogeneous weighted graph that models latent cross-cascade interactions to alleviate the first issue, capturing both local user influence patterns and global information propagation dynamics. For the second issue, contrastive learning can provide a principled framework for maintaining consistency across different-scale prediction tasks of users. Contrastive learning aims to learn representations by constructing positive and negative sample pairs, bringing positive pairs closer in the feature space while pushing negative pairs apart. Through this mechanism, we align user representations derived from the social network with those from information cascades, thereby capturing user identity invariance across these distinct contexts. Based on these, we propose a constrained cascade neighbor contrastive learning to enforce feature similarity of the same user at different scales, which can be regarded as the additional connection between two prediction tasks.

In this research, we realize the aforementioned ideas, and the novel designs constitute a new multi-scale prediction model of information diffusion by leveraging multi-dimension feature interaction, namely MDFI. First, we construct the user-cascade graph, whose nodes represent users and cascades, to capture complex hierarchical interactions in terms of user-user and user-cascade. At the user-level, it models the latent interactions between a user and strangers across different cascades; at the cascade-level, it also captures the latent interactions between a user and indirectly related cascades. Thus, it extends traditional user-only graphs [10, 13] by encompassing a wider spectrum of interactions, thus providing more comprehensive modeling of real-world social dynamics. Second, we propose cascade neighbor contrastive learning, to enforce similarity between representations of the same user at different scales, which innovatively integrates two complementary node sources, i.e., social neighbor nodes from the social network and strongly associated neighbor nodes from the user-cascade graph, thereby connecting the two tasks through maintaining the intrinsic association between them. The transformed user vectors are subsequently fed into a shared representation layer designed to extract shared features, so as to integrate the macroscopic and microscopic characteristics. Thus, this not only pulls close the representations of the same user in the feature space, but also facilitates feature interactions between macroscopic and microscopic levels.

The main contributions of this research are at least fourfold.

• We introduce a novel information diffusion prediction model that improves the performance of both macro-level and micro-level tasks through feature interaction and coordination between these two tasks.

• We propose to construct a heterogeneous weighted user-cascade graph to model comprehensive interactions, including both explicit and latent interactions between users as well as users and cascades.

• We devise cascade neighbor contrastive learning as a constraint on the similarity of user representations at different scales to better extract shared cross-task features.

• Our model is evaluated on four benchmark datasets, and experimental results indicate that MDFI consistently outperforms state-of-the-art methods on the multi-scale information diffusion prediction task.

2 Related work

This section discusses closely related work on information diffusion prediction and graph contrastive learning.

2.1 Information diffusion prediction

Current information diffusion models can be categorized into macroscopic prediction models, microscopic prediction models, and multi-scale diffusion prediction models.

2.1.1 *Macroscopic prediction*

Previous methods can be broadly classified into three categories: feature-based methods, generation-based methods, and deep learning-based methods.

Feature-based methods involve manually extracting features from data [14–16] such as topics [17] and structural patterns [18]. These methods require intricate manual feature selection and extensive domain knowledge, making them challenging to generalize to other domains. Generation-based methods model the information arrival process [19, 20]. While these methods enhance model interpretability, they often ignore the interaction information within cascades and thus negatively impact performance.

Deep learning-based methods have become dominant due to their effectiveness. For example, Li et al. [6] extracted node sequences from cascade graphs and encoded them using the gated recurrent units (GRU), a specific type of recurrent neural network (RNN). Cao et al. [21] integrated the Hawkes process with GRU, utilizing deep learning to simulate the interpretable components of the Hawkes process. Cao et al. [22] employed stacked graph neural networks (GNNs) to capture information diffusion patterns within user social networks. Chen et al. [23] sampled sub-cascade graphs from cascades and used GNNs alongside RNNs to capture the evolution of cascade structures. Zhou et al. [24] combined graph wavelets, variational autoencoders, and GRU to learn both the structural and temporal information of cascades.

2.1.2 *Microscopic prediction*

Traditional microscopic prediction methods can be mainly divided into three categories: diffusion-based methods, embedding-based methods, and deep learning-based methods.

Diffusion-based methods predict by learning the diffusion probabilities of all links in a social network [4], such as the independent cascade (IC) or linear threshold (LT) [4] models. These methods require prior assumptions about the diffusion mechanisms, making them difficult to apply to real-world scenarios. Embedding-based methods generally parameterize users [25]. For instance, Bourigault et al. [26] learnt user representations by embedding users into a continuous latent space. Zhang et al. [25] extended user parameterization to the IC model by using vectors to reflect user social influence. However, these methods are often shallow networks that struggle to effectively learn complex diffusion patterns and tend to overlook historical diffusion information.

Similar to macroscopic prediction, deep learning has been widely applied in microscopic prediction as well. Wang et al. [7] proposed using diffusion topology to describe cascade structures and extending it to long short-term memory (LSTM) networks. Islam et al. [27] combined recurrent neural networks (RNN) and attention mechanisms to model temporal information. Yang et al. [28] integrated self-attention mechanisms and convolutional neural networks (CNN) based on two heuristic assumptions. Sankar et al. [29] captured user social homophily and temporal influence by combining variational autoencoders (VAE) with co-attention mechanisms. Yuan et al. [13] modelled cascades in the form of diffusion graphs and jointly made predictions using the user social network. Sun et al. [30] further modelled cascades as diffusion hypergraphs based on this approach. Liu et al. [31] modelled the cascade by considering users as having dual roles, i.e., as both information senders and receivers.

2.1.3 *Multi-scale diffusion prediction*

Representative multi-scale diffusion prediction models includes Forest [8], DMT-LIC [9], and MINDS [10]. Forest [8] employs reinforcement learning to guide macroscopic prediction based on microscopic prediction, which lacks the recognition of the synergistic enhancements between the two tasks. DMT-LIC [9] introduces a shared representation layer to encode both cascade structure and node sequence information. However, the shared features suffer from redundancy. To address this issue, MINDS [10] incorporates adversarial learning to mitigate feature redundancy within shared features.

However, existing methods overlook the hierarchical latent interactions between users and cascades, as well as the constraint of feature similarity at different levels for the same users, leading to suboptimal performance. Here, we address these issues by constructing a user-cascade graph and proposing cascade neighbor contrastive learning.

2.2 Contrastive learning

Contrastive learning aims to learn representations by constructing positive and negative sample pairs, bringing positive pairs closer in the feature space while pushing negative pairs apart, which helps capture structural properties and invariance in the data. Currently, contrastive learning has been widely applied in various domains, e.g., image classification [32], hypergraph learning [33], and rumour detection [34].

In the field of graph learning, contrastive learning has been adopted to achieve more robust representations, with techniques like node-level [35] and edge-level [36] data augmentation. For example, Gong et al. [37] enhanced the model by modifying view encoders rather than perturbing graph inputs, while Shen et al. [38] expanded the positive sample pairs through graph neighbor contrastive learning. In information diffusion prediction, contrastive learning is employed to capture fine-grained diffusion representations within cascades [39]. In this work, we propose cascade neighbor contrastive learning as a constraint for feature similarity across different-level tasks.

3 Problem formulation

First, we describe the inputs of the model—the social network and the information cascades.

Definition 1 (Social network)**.** The social network is represented as a graph $\mathcal{G}_{\mathcal{S}} = (\mathcal{U}, \mathcal{E})$, where $\mathcal{U} = \{u_i | 1 \leqslant i \leqslant N\}$ is the set of users and $\mathcal{E} = \{e_{i,j} | 1 \leqslant i, j \leqslant N\}$ is the set of edges representing social relations between the user u_i and u_j.

Definition 2 (Information cascade)**.** An information cascade c_m is an ordered sequence of user participation for a specific piece of information. It is formally represented as a list of tuples: $c_m = \{(u_i, t_i) | u_i \in \mathcal{U}\}$, where (u_i, t_i) represents the user u_i participates the information cascade at the timestamp t_i. The sequence is sorted by time, such that the timestamps are non-decreasing.

Then, we define the tasks of macroscopic, microscopic, and multi-scale prediction, respectively.

Definition 3 (Macroscopic prediction)**.** Given a social network $\mathcal{G}_{\mathcal{S}}$, an observed diffusion cascade $c_m = \{(u_i, t_i) | u_i \in \mathcal{U}\}$, predict the final size $|c_m|$ of the cascade c_m.

Definition 4 (Microscopic prediction)**.** Given a social network $\mathcal{G}_{\mathcal{S}}$, an observed diffusion cascade $c_m = \{(u_i, t_i) | u_i \in \mathcal{U}\}$, predict the next participating user u_{i+1} at time t_{i+1}.

It follows the definition of multi-scale prediction that is a combination of macroscopic and microscopic predictions.

4 Method

In this section, we provide a detailed explanation of the model. We first describe how to capture user features from the social network and information cascades (Subsections 4.1 and 4.2), then explain how to extract private and shared features (Subsection 4.3), and finally present the diffusion prediction module (Subsection 4.4). The overall architecture, as illustrated in Figure 2, consists of four key components: (1) hierarchical feature interaction module (Subsection 4.1) learns comprehensive feature interactions between users as well as users and cascades by constructing a heterogeneous weighted user-cascade graph; (2) social feature interaction module (Subsection 4.2) captures social dependency among users leveraging the social network; (3) multi-scale task feature learning module (Subsection 4.3) focuses on learning task-specific features for both macroscopic and microscopic predictions while also capturing interactions between these features. Additionally, it learns shared features that are common across both tasks, facilitating the multi-scale learning process; (4) diffusion prediction module (Subsection 4.4) integrates

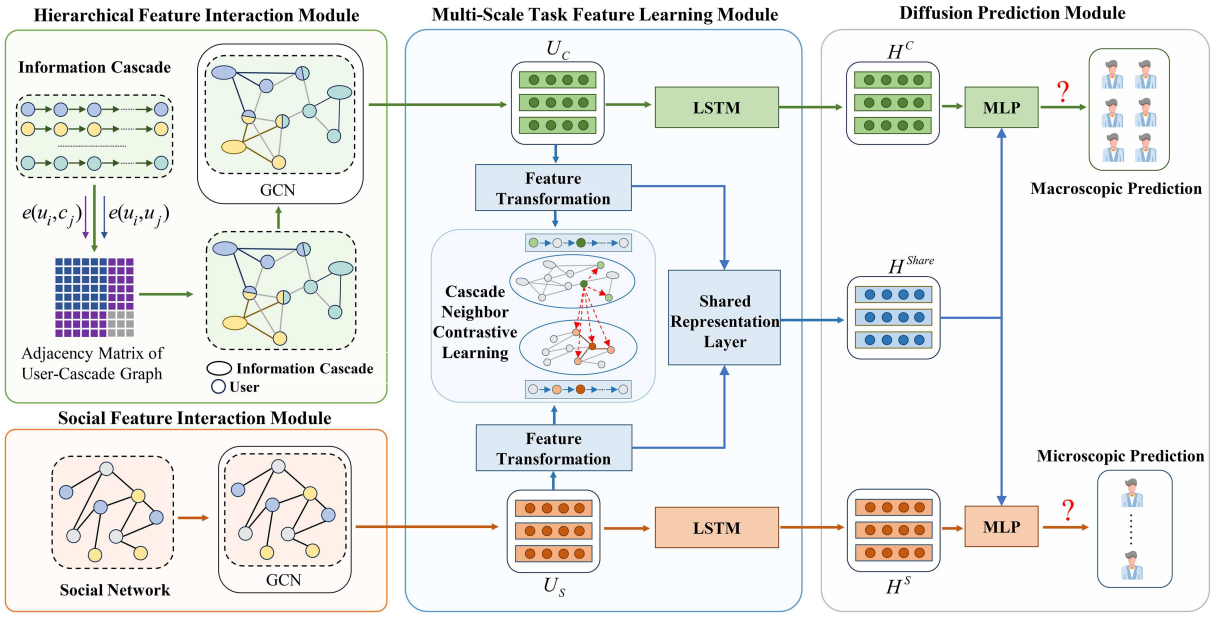

Figure 2 (Color online) The overview architecture of MDFI. Hierarchical feature interaction module and social feature interaction module learn user representations from cascades and the social network. Multi-scale task feature learning module learns task-specific and shared features of users. The diffusion prediction module makes the prediction.

task-specific features and shared features to perform multi-scale information diffusion prediction, providing a unified output that balances the influence of both macroscopic and microscopic tasks.

4.1 Hierarchical feature interaction module

To effectively model the explicit and latent interactions between users and cascades, we construct a heterogeneous weighted user-cascade graph, with nodes representing users and cascades. Subsequently, a multi-layer graph convolutional network (GCN) is employed to capture the feature representations of users, enabling the model to learn complex dependencies and interactions within the graph.

4.1.1 *User-cascade graph*

An information cascade refers to a sequence of information propagation driven by users, where users are influenced not only by the current cascade they participate in but also by other cascades they are not directly related to. We argue this concept is somewhat analogous to natural language processing (NLP) tasks, where users and cascades can be analogized to words and sentences, respectively. Inspired by the work of Yao et al. [40], we construct a heterogeneous weighted user-cascade graph $\mathcal{G}_C$ from user and cascade levels to model the explicit and latent hierarchical interactions between users and cascades.

$\mathcal{G}_C$ consists of two types of nodes: users and cascades. Thus, edges in the $\mathcal{G}_C$ are categorized into two types: user-to-user edges and user-to-cascade edges. At the user-level, the user-to-user edges are constructed based on user co-occurrence across all cascades, and at the cascade-level, the user-to-cascade edges are formed based on user participation in individual cascades. As shown in Figure 3, the user-cascade graph enables us to model latent interactions between the anchor user and indirectly related cascades as well as users within cascades. For instance, user u_4 in cascade c_2 may interact with cascade c_1 or u_2 through u_1, representing the latent interactions.

To effectively model global user co-occurrence, we apply a sliding window to gather co-occurrence statistics. We then utilize positive point-wise mutual information (PPMI) to calculate the weights of edges between users, with the computation formula as follows:

$$\mathrm{PPMI}(u_i,u_j)=\max\left\{0,\log\frac{p(u_i,u_j)}{p(u_i)p(u_j)}\right\}, \tag{1}$$

$$p(u_i,u_j)=\frac{W(u_i,u_j)}{W}, \tag{2}$$

$$p(u_i)=\frac{W(u_i)}{W}, \tag{3}$$

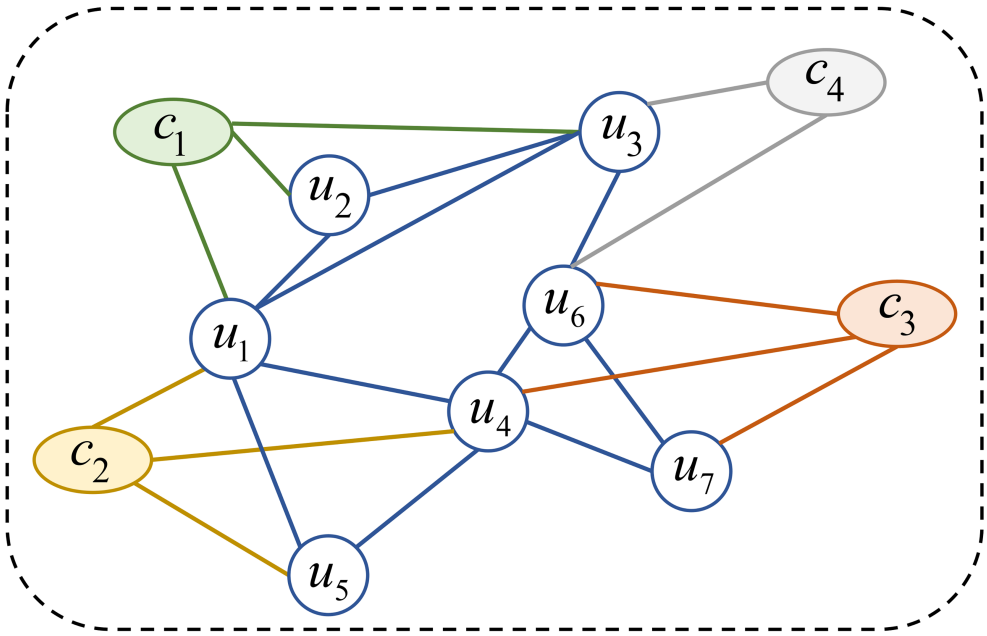

Figure 3 (Color online) Illustration of the user-cascade graph. u_i and c_i represent the user and cascade, respectively.

where $W(u_i)$ is the number of sliding windows that contain user u_i, $W(u_i, u_j)$ is the number of sliding windows that contain both user u_i and u_j, and W is the number of sliding windows. A higher PPMI value indicates a high correlation between users.

To compute the weights between users and cascades, we apply the term frequency-inverse document frequency (TF-IDF) of the user in the cascade. In this context, the term frequency represents how many times a user appears within a cascade, while inverse document frequency is defined as the logarithmic inverse of the number of cascades in which the user participates. The TF-IDF formula is as follows:

$$\text{TF-IDF}(u_i, c_j) = \text{TF}(u_i, c_j) \cdot \text{IDF}(u_i), \tag{4}$$

$$\text{TF}(u_i, c_j) = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{5}$$

$$\text{IDF}(u_i) = \log \frac{|C|}{1 + |j : u_i \in c_j|}, \tag{6}$$

where $n_{i,j}$ is the times that user u_i appears in the cascade c_j, $|C|$ is the number of cascades, $|j : u_i \in c_j|$ is the number of cascades containing user u_i. The higher the TF-IDF value of one user, the higher the importance of the user.

4.1.2 *User hierarchical representation learning*

Once the user-cascade graph is constructed, we employ a multi-layer GCN to derive the user representations. Given a user-cascade graph $\mathcal{G}_C$, the propagation rule for each layer is defined as

$$\boldsymbol{U}_C^{l+1} = \sigma\left(\tilde{\boldsymbol{D}}_C^{-\frac{1}{2}} \tilde{\boldsymbol{A}}_C \tilde{\boldsymbol{D}}_C^{-\frac{1}{2}} \boldsymbol{U}_C^l \boldsymbol{W}_C\right), \tag{7}$$

where $\boldsymbol{W}_C$ is the trainable weight matrix, $\tilde{\boldsymbol{A}}_C = \boldsymbol{A}_C + \boldsymbol{I}$, $\boldsymbol{I}$ refers to the identity matrix, $\tilde{\boldsymbol{A}}_C$ and $\tilde{\boldsymbol{D}}_C$ are the adjacency and degree matrices of self-looped $\mathcal{G}_C$. $\sigma(\cdot)$ is the ReLU activation function. The $\boldsymbol{U}_C^{(0)} \in \mathbb{R}^{(N+M)\times d}$ is the initial user embedding matrix that is randomly initialized from the normal distribution, N is the number of users, M is the number of cascades, d is the dimension of embedding. We can obtain the final user hierarchical representation $\boldsymbol{U}_C$ through several layers of GCN.

4.2 Social feature interaction module

Social homophily implies that users typically engage in interactions with others who share common traits or interests, which can be reflected in social networks. To model these social connections, we leverage the user social network to

capture their social relations and embed the homophily through a multi-layer GCN. Given the user social network $\mathcal{G}_S$, the feature matrix for users at layer $L+1$ is computed as follows:

$$\boldsymbol{U}_S^{l+1} = \sigma\left(\tilde{\boldsymbol{D}}_S^{-\frac{1}{2}}\tilde{\boldsymbol{A}}_S\tilde{\boldsymbol{D}}_S^{-\frac{1}{2}}\boldsymbol{U}_S^{l}\boldsymbol{W}_S\right), \tag{8}$$

where $\boldsymbol{W}_S$ is the trainable weight matrix, $\tilde{\boldsymbol{A}}_S = \boldsymbol{A}_S + \boldsymbol{I}$, $\boldsymbol{I}$ refers to the identity matrix, $\tilde{\boldsymbol{A}}_S$ and $\tilde{\boldsymbol{D}}_S$ are the adjacency and degree matrices of self-looped $\mathcal{G}_S$. $\sigma(\cdot)$ is the ReLU activation function. The $\boldsymbol{U}_S^{(0)} \in \mathbb{R}^{N\times d}$ is the initial user embedding matrix that is randomly initialized from the normal distribution, N is the number of users, d is the dimension of embedding. We can obtain the final user social representation $\boldsymbol{U}_S$ through several layers of GCN.

4.3 Multi-scale task feature learning module

Graph-based representation learning captures both hierarchical feature interactions between users and cascades as well as their social homophily interactions. However, it is limited in modeling the contextual interactions within cascades. To address this, we use LSTMs to capture both the social and hierarchical contextual interactions of users within cascades, which is referred to as task-specific feature learning.

4.3.1 *Task-specific representation learning*

Given user hierarchical representation $\boldsymbol{U}_C$ and social representation $\boldsymbol{U}_S$, we generate user sequence representations according to the order of users in the cascades. Subsequently, we utilize two LSTM modules to compute the representation of contextual interactions for them as follows:

$$\boldsymbol{h}_t^C = \mathbf{LSTM}(\boldsymbol{h}_{t-1}^C, \boldsymbol{u}_t^C, \theta_C), \tag{9}$$

$$\boldsymbol{h}_t^S = \mathbf{LSTM}(\boldsymbol{h}_{t-1}^S, \boldsymbol{u}_t^S, \theta_S), \tag{10}$$

where $\boldsymbol{h}_t^C$ and $\boldsymbol{h}_t^S$ are the hidden states. θ_C and θ_S are parameters in LSTM. Thus we can obtain the task-specific representations $\boldsymbol{H}^C \in \mathbb{R}^{N\times d}$ and $\boldsymbol{H}^S \in \mathbb{R}^{N\times d}$.

We observe that features of a user exhibit a certain degree of similarity across macroscopic and microscopic prediction tasks, which can be leveraged within a multi-task learning framework to strengthen the connection between the two tasks. Hence, we propose cascade neighbor contrastive learning as a constraint on feature similarity for the same user across both tasks. Additionally, there is the existence of hidden shared features that can benefit both macroscopic and microscopic tasks [10]. To capture these, we develop a shared representation layer based on LSTM and GRU.

4.3.2 *Cascade neighbor contrastive learning*

Inspired by recent advancements in graph contrastive learning [37, 38], we employ contrastive learning to enforce feature similarity between macro-level and micro-level representations. Considering that user social relations and retweeting behaviors are mutually influential, we propose cascade neighbor contrastive learning, which simultaneously takes into account both the social relations within the social network and the strong connections in the user-cascade graph.

The neighbor nodes in a cascade can be categorized into two types: (1) social neighbor nodes from the user social network, and (2) strongly associated neighbor nodes from the user-cascade graph. For the former, we generate the cascade adjacency matrix $\boldsymbol{A}_{cas}^S$ for the cascade by the adjacency matrix $\boldsymbol{A}_S$, which reflects the user social relations in the cascade. For the latter, edge weights between users in the user-cascade graph $\mathcal{G}_C$ above the mode value are considered as the strong association, and then we generate the cascade adjacency matrix $\boldsymbol{A}_{cas}^C$ from these strongly associated nodes.

Given a cascade $c_m = \{(u_i, t_i)|u_i \in \mathcal{U}\}$, we compute its sequence representations $\boldsymbol{c}_m^C = [(\boldsymbol{u}_i^C)] \in \mathbb{R}^{|c_m|\times d}$ and $\boldsymbol{c}_m^S = [(\boldsymbol{u}_i^S)] \in \mathbb{R}^{|c_m|\times d}$, representing cascade view 1 (user hierarchical representations) and cascade view 2 (user social representations), respectively. Next, we pass the sequence representations $\boldsymbol{c}_m^C$ and $\boldsymbol{c}_m^S$ through the feature transformation (FT) module to project them into the same level of feature space. The formula is as follows:

$$\boldsymbol{c}_m^{(1)}, \boldsymbol{c}_m^{(2)} = \mathrm{FT}(\boldsymbol{c}_m^C, \boldsymbol{c}_m^S), \tag{11}$$

where

$$\mathrm{FT}(\boldsymbol{x}) = \mathrm{DN}((\sigma(\boldsymbol{x}\boldsymbol{W}_1^{\mathrm{FT}} + \boldsymbol{b}_1^{\mathrm{FT}})\boldsymbol{W}_2^{\mathrm{FT}} + \boldsymbol{b}_2^{\mathrm{FT}}) + \boldsymbol{x}), \tag{12}$$

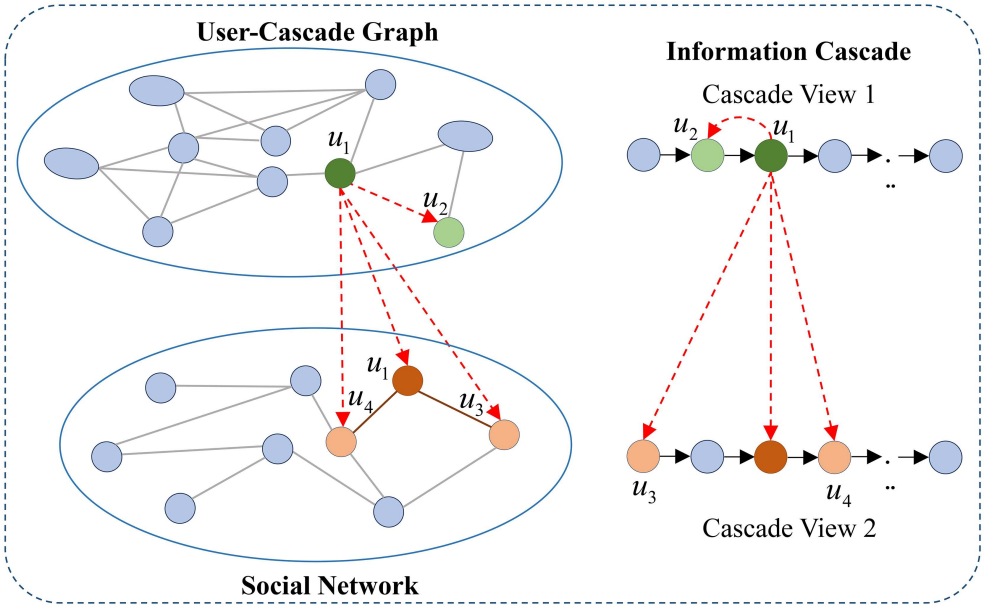

Figure 4 (Color online) Illustration of cascade neighbor contrastive learning. For u_1, u_2 is the strongly associated neighbor user, and u_3, u_4 are the social neighbor users.

and $\boldsymbol{W}^{\mathrm{FT}}$ and $\boldsymbol{b}^{\mathrm{FT}}$ are the weight matrix and bias, respectively; $\mathrm{DN}(\cdot)$ is the Dropout and normalization function.

Thus, let $\boldsymbol{u}_i^{(1)}$ and $\boldsymbol{u}_i^{(2)}$ denote the embeddings of node u_i within cascade views 1 and 2, the positive sample pairs can be classified into five types: (1) the same node across different cascade views, i.e., $(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_i^{(2)})$; (2) strongly associated neighbor nodes within cascade view 1, i.e., $\{(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_j^{(1)})|u_j \in N_i^C\}$; (3) social neighbor nodes within cascade view 2, i.e., $\{(\boldsymbol{u}_i^{(2)}, \boldsymbol{u}_j^{(2)})|u_j \in N_i^S\}$; (4) strongly associated neighbor nodes across cascade views, i.e., $\{(\boldsymbol{u}_i^{(2)}, \boldsymbol{u}_j^{(1)})|u_j^{(1)} \in N_i^C\}$; and (5) social neighbor nodes across cascade views, i.e., $\{(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_j^{(2)})|u_j^{(2)} \in N_i^S\}$. Other node pairs are the negative sample pairs. An example based on cascade view 1 is shown in Figure 4. In this case, u_2 is the strongly associated neighbor user of u_1, while u_3 and u_4 are its social neighbor users, making them positive examples, whereas all other users in the cascade serve as negative examples.

Below is the contrastive objective in cascade view 1:

$$\ell(\boldsymbol{u}_i^{(1)}) = -\log \frac{\left(\mathrm{e}^{\theta(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_i^{(2)})/\tau} + \ell_p^1\right)/(|N_i^S| + |N_i^C| + 1)}{\mathrm{e}^{\theta(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_i^{(2)})/\tau} + \ell_N}, \tag{13}$$

$$\ell_N = \sum_{j \neq i} \left(\mathrm{e}^{\theta(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_j^{(1)})/\tau} + \mathrm{e}^{\theta(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_j^{(2)})/\tau}\right), \tag{14}$$

$$\ell_p^1 = \sum_{u_j \in N_i^C} \mathrm{e}^{\theta(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_j^{(1)})/\tau} + \sum_{u_j \in N_i^S} \mathrm{e}^{\theta(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_j^{(2)})/\tau}, \tag{15}$$

where $\theta(\cdot)$ is the function that measures the similarity, N_i^C is the strongly associated neighbor users of the user u_i, N_i^S is the social neighbor users of the user u_i, ℓ_p^1 is positive samples in the intra cascade view and inter cascade view, and ℓ_N is the negative samples in two views. Unlike traditional contrastive learning, the contrastive objectives in cascade views 1 and 2 are not identical. The contrastive objective in cascade view 2 $\ell(\boldsymbol{u}_i^{(2)})$ differs from that in $\ell(\boldsymbol{u}_i^{(1)})$, primarily due to differences in the definition of ℓ_p^2, which is formulated as

$$\ell_p^2 = \sum_{u_j \in N_i^S} \mathrm{e}^{\theta\left(\boldsymbol{u}_i^{(2)}, \boldsymbol{u}_j^{(2)}\right)/\tau} + \sum_{u_j \in N_i^C} \mathrm{e}^{\theta\left(\boldsymbol{u}_i^{(2)}, \boldsymbol{u}_j^{(1)}\right)/\tau}. \tag{16}$$

Thus, the final contrastive loss is defined as

$$\mathcal{L}_{ncl} = \frac{1}{2M} \sum_{i=1}^{M} \left[\ell\left(u_i^{(1)}\right) + \ell\left(u_i^{(2)}\right)\right], \tag{17}$$

where M is the number of cascades.

Finally, we fuse the cascade sequence representations constrained by contrastive learning with the original cascade sequence representations to form the new cascade sequence representation. The formula is as

$$\boldsymbol{c}_m^C, \boldsymbol{c}_m^S = \alpha \cdot (\boldsymbol{c}_m^C, \boldsymbol{c}_m^S) + (1-\alpha) \cdot (\boldsymbol{c}_m^{(1)}, \boldsymbol{c}_m^{(2)}), \tag{18}$$

where α is the fusion coefficient.

4.3.3 *Shared representation learning*

We design a novel shared representation layer (SRL) to capture shared features based on LSTM and GRU. We adopt this gating architecture to function as an efficient and adaptive filter for information fusion. This design allows the model to dynamically modulate the information flow, for instance, by discarding irrelevant historical noise via the forgetting logic. This ensures a direct and lightweight fusion process for extracting the shared representation. The layer is defined as

$$\boldsymbol{i}_t = \tanh(\boldsymbol{u}_{t-1}^C \boldsymbol{Q}_i + \boldsymbol{u}_{t-1}^S \boldsymbol{K}_i + h_{t-1} \boldsymbol{V}_i + \boldsymbol{b}_i), \tag{19}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{u}_{t-1}^C \boldsymbol{Q}_f + \boldsymbol{u}_{t-1}^S \boldsymbol{K}_f + \boldsymbol{h}_{t-1} \boldsymbol{V}_f + \boldsymbol{b}_f), \tag{20}$$

$$\boldsymbol{z}_t = \sigma(\boldsymbol{u}_{t-1}^C \boldsymbol{Q}_z + \boldsymbol{u}_{t-1}^S \boldsymbol{K}_z + \boldsymbol{c}_{t-1} \boldsymbol{V}_z + \boldsymbol{b}_z), \tag{21}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \cdot \boldsymbol{c}_{t-1} + (1 - \boldsymbol{f}_t) \cdot \boldsymbol{i}_t, \tag{22}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \cdot \tanh(\boldsymbol{c}_t) + (1 - \boldsymbol{z}_t) \cdot \boldsymbol{h}_{t-1}, \tag{23}$$

where $\boldsymbol{Q}_* \in \mathbb{R}^{d \times d}$, $\boldsymbol{K}_* \in \mathbb{R}^{d \times d}$, $\boldsymbol{V}_* \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b}_* \in \mathbb{R}^{d \times d}$ are trainable parameters. The input gate $\boldsymbol{i}_t$ controls the information that needs to be input, while the forget gate $\boldsymbol{f}_t$ is responsible for discarding irrelevant information from the past and works together with the input gate to update the current cell state $\boldsymbol{c}_t$. The reset gate $\boldsymbol{z}_t$ is used to control the influence of $\boldsymbol{h}_{t-1}$ and, along with the $\boldsymbol{c}_t$, computes the output state $\boldsymbol{h}_t$. Through the shared representation layer, we can obtain the shared representation, denoted as $\boldsymbol{H}^{share}$.

4.4 Diffusion prediction module

To leverage both task-specific and shared features for enhanced prediction accuracy, we concatenate $\boldsymbol{H}^C$ and $\boldsymbol{H}^S$ with $\boldsymbol{H}^{share}$ respectively, and then feed them into different output layers to make macroscopic and microscopic predictions.

4.4.1 *Macroscopic prediction*

We aim to predict the final size of the cascade, which refers to the total number of users retweeting the information within the cascade over time. The calculation of the final size is as follows:

$$\boldsymbol{S}_m = \mathrm{MLP}(\mathrm{concat}(\boldsymbol{h}^C, \boldsymbol{h}^{share})), \tag{24}$$

where $\mathrm{concat}(\cdot, \cdot)$ is the concatenation operation. We adopt the following loss function for training:

$$\mathcal{L}_{macro} = \frac{1}{M} \sum_{m=1}^{M} (\boldsymbol{S}_m - \hat{\boldsymbol{S}}_m), \tag{25}$$

where M is the number of cascades and $\hat{\boldsymbol{S}}_m$ is the ground truth.

4.4.2 *Microscopic prediction*

We aim to predict the participating probability $\boldsymbol{p}_i \in \mathbb{R}^{|c_m|}$ of the user u_i, i.e.,

$$\boldsymbol{p}_i = \mathrm{softmax}(\mathrm{MLP}(\mathrm{concat}(\boldsymbol{h}^S, \boldsymbol{h}^{share}))). \tag{26}$$

We adopt the cross-entropy loss for training, i.e.,

$$\mathcal{L}_{micro} = -\sum_{j=2}^{|c_m|} \sum_{i=1}^{N} \hat{p}_{ji} \log(\boldsymbol{p}_{ji}), \tag{27}$$

where $\hat{p}$ is the ground truth. If user u_i participate in the cascade c_m in step j, then $\hat{p}_{ji} = 1$, otherwise $\hat{p}_{ji} = 0$.

The overall loss function of our model is defined as

$$\mathcal{L} = \lambda \mathcal{L}_{macro} + (1 - \lambda)\mathcal{L}_{micro} + \beta \mathcal{L}_{ncl}, \tag{28}$$

where λ is the balance coefficient and β is the hyperparameter.

5 Experiments

We conduct extensive experiments on four real-world datasets. The experiments were designed to address the following key research questions.

- **RQ1:** How does our model MDFI perform compared with the baselines on two tasks?
- **RQ2:** What is the impact of hyperparameter settings on the performance of MDFI?
- **RQ3:** How do specific components contribute to the overall effectiveness of MDFI?
- **RQ4:** To what extent do user cascade graphs and cascade neighbor contrastive learning improve the performance of MDFI?
- **RQ5:** How do social neighbor nodes and strongly associated neighbor nodes in contrastive learning affect the performance of MDFI?
- **RQ6:** How does the efficiency of MDFI compare to the baseline MINDS?

5.1 Experimental setup

5.1.1 *Datasets*

Following the previous work [10], we conduct experiments on four datasets: `Christianity`, `Android`, `Douban`, and `Memetracker`. The statistics are in Table 1. We randomly divide them into train/valid/test sets with an 80/10/10 ratio. Specifically, the `Christianity` [29] dataset is collected from a community Q&A website called Stack-Exchange, which is related to the Christian topic. The social network is formed through user interactions (e.g., comments, upvotes, etc.). The `Android` [29] dataset is similar to the `Christianity` dataset, which focuses on discussion around the Android topic. The `Douban` [41] dataset is collected from a social website where users can share their reading or movie-watching statuses and follow others' statuses. The `Memetracker` [42] dataset consists of over a million news articles and blog posts and tracks the most common memes. Each meme is treated as a piece of information, and each URL is considered a user.

5.1.2 *Baseline models*

We compared our model with nine baseline models, which can be classified into three categories: (1) macroscopic prediction models, (2) microscopic prediction models, and (3) unified multi-scale prediction models.

Macroscopic prediction model. (1) TCSE-net [43] employs LSTM and GCN to learn the trend representation and the cascade representation, respectively. These two representations are then aligned in time sequence and fused for prediction. (2) CasFlow [44] models the diffusion process by learning both structural and temporal information, capturing the uncertainty of the cascade representation and the node infection and achieving hierarchical pattern learning.

Microscopic prediction model. (1) TopoLSTM [7] uses diffusion topology to describe the cascade structure and extends the LSTM model for prediction. (2) NDM [28] is based on two assumptions and integrates self-attention with CNN to make predictions. (3) DyHGCN [13] jointly learns the social network and diffusion relations and incorporates temporal information to capture the dynamic preferences of users. (4) TAN-DRUD [31] models users as information senders and receivers to capture user dependencies.

Unified multi-scale prediction model. (1) Forest utilizes an RNN-based framework for microscopic prediction and incorporates reinforcement learning to guide macroscopic prediction using the microscopic model. (2) DMT-LIC [9] designs a shared representation layer to encode the cascade structure and node sequence information jointly. (3) MINDS [10] leverages the shared LSTM to learn a hidden common feature between two tasks and introduces adversarial training to mitigate feature redundancy.

5.1.3 *Evaluation metrics*

Following the previous work [10], we use mean squared logarithmic error (MSLE) as the evaluation metric for macroscopic prediction. For microscopic prediction, we use the hit scores on top k (Hits@k) and mean average precision on top k (MAP@k) as the evaluation metric, where $k = [10, 50, 100]$.

Table 1 Statistics of datasets.

Dataset	Christianity	Android	Douban	Memetracker
#Users	2897	9958	12232	4709
#Links	35624	48573	396580	209194
#Cascades	589	679	3475	12661
Average length	22.9	33.3	21.76	1624

Table 2 Experimental results of microscopic prediction on four datasets (%) (Hits@k scores for k = 10, 50, 100). The best results are in bold font. * indicates that the improvement of MDFI over MINDS is statistically significant.

Model	Christianity			Android			Douban			Memetracker		
	@10	@50	@100	@10	@50	@100	@10	@50	@100	@10	@50	@100
TopoLSTM	15.59	36.53	47.77	4.60	13.18	21.03	3.06	1.43	1.84	19.08	36.87	46.83
NDM	4.64	11.45	14.61	1.70	4.23	5.55	3.88	5.06	5.28	9.31	12.28	12.79
DyHGCN	23.80	46.89	59.23	7.48	17.46	25.96	14.38	26.48	33.29	25.22	46.03	57.10
TAN-DRUD	19.08	44.06	56.97	2.81	10.24	16.58	8.41	16.04	21.75	21.39	42.47	53.83
Forest	27.01	47.32	57.37	8.27	16.03	21.64	16.36	28.77	35.92	28.27	47.21	57.17
DMT-LIC	25.45	45.75	56.47	8.64	17.13	22.90	19.27	31.25	37.17	27.20	47.75	58.69
MINDS	30.88	50.94	61.25	9.94	19.48	26.15	19.56	32.48	39.76	27.69	48.76	59.72
MDFI	**32.12***	**52.12***	**63.26***	**10.31***	**20.33***	**27.91***	**21.37***	**33.06***	**40.55***	**29.22***	**50.10***	**60.65***

5.1.4 *Model configurations*

For baseline models, all the experimental results are cited from [10]. For Forest, DMT-LIC, CasFlow, and MINDS, we follow the parameter settings in the resource codes. For MDFI, we set the batch size to 32, the embedding dimension to 64, and perform a parameter search on several hyperparameters.

5.2 Performance comparison (RQ1)

We compare MDFI with baseline models on four public datasets, with results for microscopic prediction shown in Tables 2 and 3, and for macroscopic prediction in Table 4. From the experiments, we draw the following findings. (1) MDFI outperforms all baseline models in the microscopic prediction task. Unlike methods focusing solely on microscopic prediction, MDFI comprehensively models interactions between users and between users and cascades. The introduction of contrastive learning effectively captures the mutual influence between user social networks and information cascades, leading to significant improvements in the performance of MDFI. (2) MDFI also excels in the macroscopic prediction task, surpassing all baseline models that specialize in macroscopic prediction in the vast majority of indicators. By integrating two levels of prediction, MDFI enables them to collaborate and enhances performance. (3) MDFI demonstrates significantly better results in multi-scale prediction tasks compared to all the baseline models. Furthermore, we conduct statistical significance tests to rigorously verify whether the improvements of MDFI over MINDS are statistically significant. Results of p_{value} confirm that our performance gains are indeed statistically significant ($p_{value} < 0.05$), indicating that the observed advantages are stable. The improvement lies in our more comprehensive modeling of the interaction between users and cascades. Moreover, by using contrastive learning as a constraint, MDFI ensures that features of the same user across different tasks exhibit a degree of similarity, which also strengthens the interaction between the social network and information cascades.

5.3 Hyperparameter sensitivity analysis (RQ2)

Using the Android dataset, we study the impact of different hyperparameter configurations on model performance. Specifically, we explore the sensitivity of the balance coefficient λ, the fusion coefficient α, and the hyperparameter coefficient β.

5.3.1 *Size of the sliding window w*

In Subsection 4.1, we utilize w as the size of the sliding window. Based on Figures 5(a) and (b), it can be observed that as the size of the sliding window w gradually increases, the performance of the model does not fluctuate significantly. Therefore, the model exhibits robustness with respect to this hyperparameter.

Table 3 Experimental results of microscopic prediction on four datasets (%) (Map@k scores for k = 10, 50, 100). The best results are in bold font. * indicates that the improvement of MDFI over MINDS is statistically significant.

Model	Christianity			Android			Douban			Memetracker		
	@10	@50	@100	@10	@50	@100	@10	@50	@100	@10	@50	@100
TopoLSTM	5.23	6.19	6.35	1.66	2.02	2.13	3.54	8.24	8.84	8.70	9.55	9.69
NDM	1.44	1.77	1.82	0.59	0.70	0.72	1.41	8.24	8.84	4.63	4.80	4.81
DyHGCN	10.62	11.67	11.84	3.92	4.34	4.46	8.01	8.56	8.65	14.10	15.02	15.18
TAN-DRUD	7.52	11.67	11.84	0.99	1.30	1.39	3.59	4.01	4.09	9.91	10.86	11.02
Forest	17.06	18.06	18.20	5.80	6.17	6.26	10.23	10.81	10.92	15.28	16.14	16.29
DMT-LIC	17.26	18.12	18.27	5.95	6.30	6.38	11.04	11.62	11.70	14.81	15.75	15.90
MINDS	18.61	19.48	19.63	6.20	6.62	6.72	10.87	11.48	11.58	14.90	15.87	16.02
MDFI	**19.38***	**20.23***	**20.39***	**6.35***	**6.79***	**6.90***	**12.17***	**12.71***	**12.82***	**15.60***	**16.59***	**16.69***

Table 4 Experimental results of macroscopic prediction on four datasets in terms of MSLE. The best results are in bold font. * indicates that the improvement of MDFI over MINDS is statistically significant.

Model	Christianity	Android	Douban	Memetracker
CasFlow	0.520	1.080	0.979	0.756
TCSE-net	2.391	2.882	1.033	2.285
Forest	0.677	0.775	0.960	**0.655**
DMT-LIC	0.547	1.366	0.984	0.811
MINDS	0.366	0.775	0.969	0.777
MDFI	**0.182***	**0.734***	**0.922***	0.745*

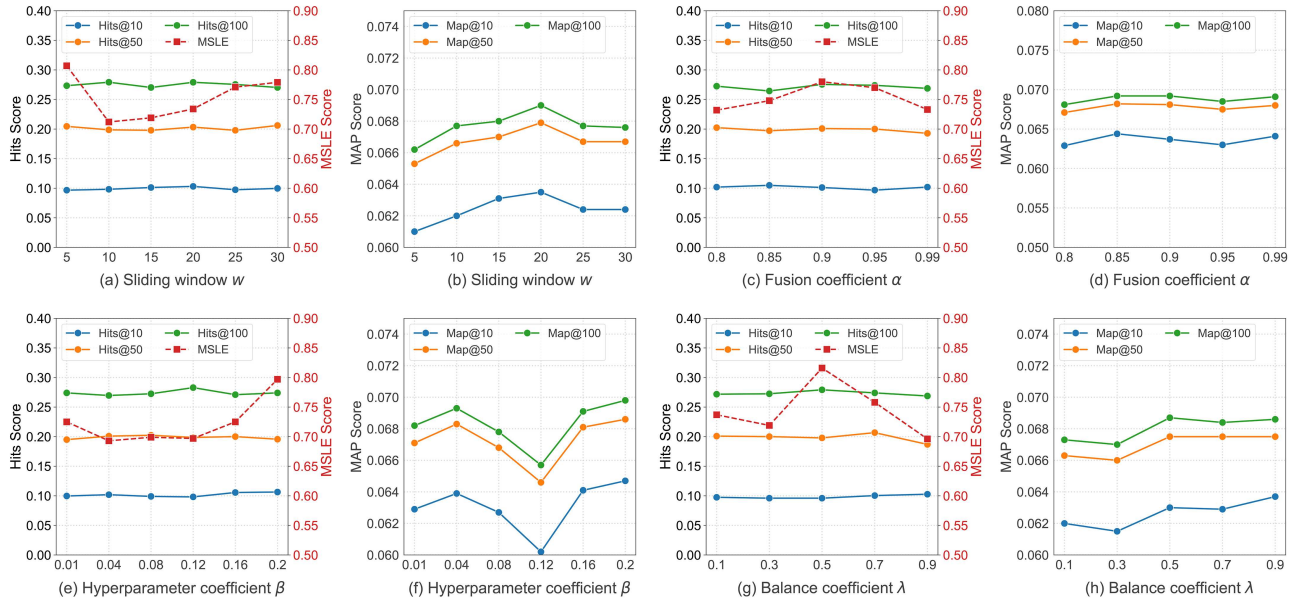

Figure 5 (Color online) Hyperparameter sensitivity analysis on the dataset. (a) and (b) illustrate the impact of sliding window size w; (c) and (d) display the effect of fusion coefficient α; (e) and (f) demonstrate the sensitivity of hyperparameter coefficient β; (g) and (h) show the influence of balance coefficient λ.

5.3.2 *Fusion coefficient* α

In Subsection 4.3, we apply α as the fusion coefficient between the cascade sequence representation constrained by contrastive learning and the original sequence representation. Illustrated by Figures 5(c) and (d), we can find that the MDFI performance shows a declining trend as α increases, which suggests that ensuring feature similarity for the same user across different-level tasks is beneficial for improving performance. The balance between these two factors is crucial for optimal performance.

Table 5 Ablation study on both microscopic (Hits@k (%) scores for $k = 10$, 100) and macroscopic (MSLE) prediction. The best results are in bold font.

Model	Christianity			Android			Douban			Memetracker		
	@10	@100	MSLE	@10	@100	MSLE	@10	@100	MSLE	@10	@100	MSLE
MDFI	**32.12**	63.26	**0.182**	**10.31**	**27.91**	**0.734**	**21.37**	**40.55**	**0.922**	**29.22**	**60.65**	0.745
w/o UC	31.07	60.94	0.213	9.82	27.35	0.970	20.91	40.31	1.051	28.74	59.39	**0.738**
w/o CACL	31.96	**63.39**	0.197	10.09	27.51	0.836	20.45	40.48	0.959	28.38	60.15	0.769
w/o Macro	30.89	63.21	9.223	9.94	27.15	15.62	20.96	40.42	14.83	28.86	59.69	14.01
w/o Micro	0.312	9.912	0.209	0.441	3.43	1.194	0.028	0.537	1.105	0.530	2.056	0.787

Table 6 Ablation study on both microscopic (Map@k (%) scores for $k = 10$, 100) and macroscopic (MSLE) prediction. The best results are in bold font.

Model	Christianity			Android			Douban			Memetracker		
	@10	@100	MSLE	@10	@100	MSLE	@10	@100	MSLE	@10	@100	MSLE
MDFI	**19.38**	**20.39**	**0.182**	**6.35**	**6.90**	**0.734**	**12.17**	**12.82**	**0.922**	**15.60**	**16.69**	0.745
w/o UC	18.58	19.64	0.213	6.22	6.77	0.970	11.92	12.59	1.051	15.35	16.42	**0.738**
w/o CACL	18.43	19.51	0.197	6.26	6.79	0.836	11.80	12.50	0.959	15.21	16.34	0.769
w/o Macro	18.75	19.87	9.223	6.25	6.78	15.62	11.99	12.67	14.83	15.44	16.54	14.01
w/o Micro	0.032	0.167	0.209	0.80	0.15	1.194	0.006	0.016	1.105	0.187	0.227	0.787

5.3.3 *Hyperparameter coefficient β*

In Subsection 4.4, we utilize β as the coefficient of the contrastive loss function. Based on Figures 5(e) and (f), we can see that the performance of the model remains relatively stable as β changes. Therefore, we can conclude that the model demonstrates robustness to variations in the β parameter.

5.3.4 *Balance coefficient λ*

In Subsection 4.4, we employ λ as the balance coefficient between macroscopic and microscopic prediction tasks. Referring to Figures 5(g) and (h), we observe that as the value of λ increases, the hits metric remains relatively stable and the map score exhibits slight fluctuations. Overall, the model performs better when the macroscopic prediction task weighting is higher.

5.4 Ablation study (RQ3)

To validate the contribution of each sub-module in MDFI, we conducted an ablation study on all four datasets. The variants are designed as follows:

- **w/o UC** removes the user-cascade graph module;
- **w/o CACL** removes cascade neighbor contrastive learning module;
- **w/o Macro** removes the macroscopic prediction part;
- **w/o Micro** removes the microscopic prediction part.

As shown in Tables 5 and 6, where we compare the performance of different variants of MDFI on both microscopic and macroscopic predictions, we arrive at the following findings. (1) Removing the user-cascade graph negatively affects the performance of MDFI, indicating that our method effectively models the interactions between users as well as users and cascades, even when one user does not have a direct relation with a cascade. (2) The performance of MDFI decreases when contrastive learning is removed, which demonstrates the effectiveness of contrastive learning in enforcing the similarity constraint on the features of the same user across different levels of prediction tasks, while also enhancing the interaction between the social network and information cascades. (3) By modeling user propagation behavior, macroscopic prediction improves microscopic predictions. Conversely, microscopic prediction enhances the macroscopic prediction of overall diffusion trends by considering social homophily as a potential propagation trend. The performance decline observed when macroscopic and microscopic predictions are removed highlights the mutual enhancement effect between the two tasks.

5.5 Performance of the user-cascade graph and cascade neighbor contrastive learning analysis (RQ4)

To evaluate the effectiveness of the user-cascade graph and cascade neighbor contrastive learning, we perform comparative experiments against the corresponding modules in MINDS on the `Christianity` and `Memetracker` datasets. MDFI(SH) is the model variant in which the user-cascade graph is replaced by sequential hypergraphs.

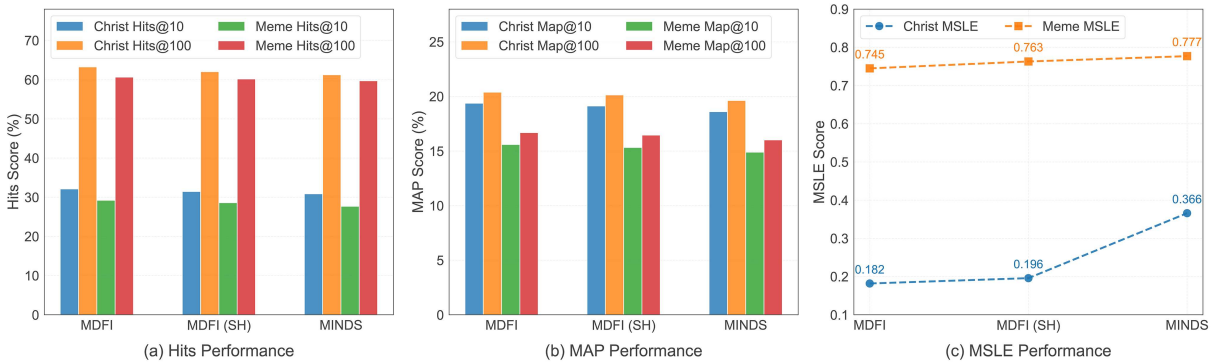

Figure 6 (Color online) To explore the effectiveness of user cascade graph and cascade neighbor contrastive learning, we conduct validation experiments on the `Christianity` and `Memetracker` datasets. (a) Comparison of three models on metrics Hits@10 and Hits@100; (b) comparison of three models on metrics Map@10 and Map@100; (c) comparison of three models on metric MSLE.

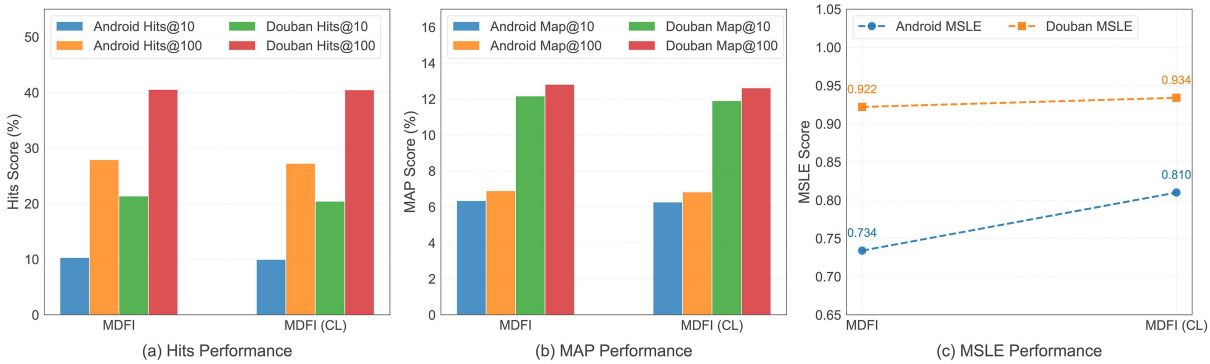

Figure 7 (Color online) To investigate the differences between cascaded neighbor contrastive learning and traditional contrastive learning, we conduct validation experiments on the `Android` and `Douban` datasets. (a) Comparison of two models on metrics Hits@10 and Hits@100; (b) comparison of two models on metrics Map@10 and Map@100; (c) comparison of two models on metric MSLE.

MDFI(SH) incorporates a cascade contrastive learning module, which is absent in MINDS. In comparison, MDFI replaces the sequential hypergraphs in MDFI(SH) with user-cascade graphs. From the results presented in Figure 6, we can observe that incorporating cascade neighbor contrastive learning improves the performance of MDFI(SH) compared to MINDS, which suggests that applying contrastive learning as a constraint to limit the similarity of the same user across different tasks is both reasonable and effective. Furthermore, by replacing the sequential hypergraphs with the user-cascade graph, MDFI outperforms MDFI(SH) on all metrics, indicating that the user-cascade graph can capture not only explicit interactions between users and between users and cascades but also implicit interactions. Overall, both the user-cascade graph and cascade neighbor contrastive learning significantly enhance the performance of MDFI.

5.6 Cascade neighbor contrastive learning analysis (RQ5)

To investigate the impact of social neighbor users and strongly associated neighbor users in contrastive learning on MDFI performance, we conduct the contrastive learning analysis on `Android` and `Douban` datasets. In this case, MDFI(CL) refers to the model variant where neither social neighbor nor strongly associated neighbor users are considered, meaning that the positive example pairs are only the same nodes across different views. As shown in Figure 7, excluding social and strongly associated neighbor users results in a noticeable drop in model performance. This finding suggests that incorporating both types of neighbor users is crucial. Social relations in the network reflect feature similarities between users, which can be explained by social homophily. In the user-cascade graph, strongly associated neighbor users are those who frequently appear in the same cascade, which also indicates that they share similar interests and preferences.

5.7 Time efficiency analysis (RQ6)

To validate the operational efficiency of MDFI, we conduct an efficiency analysis across all four datasets. Specifically, we measure the average training time (per epoch) under different values of the batchsize. For comparison, we

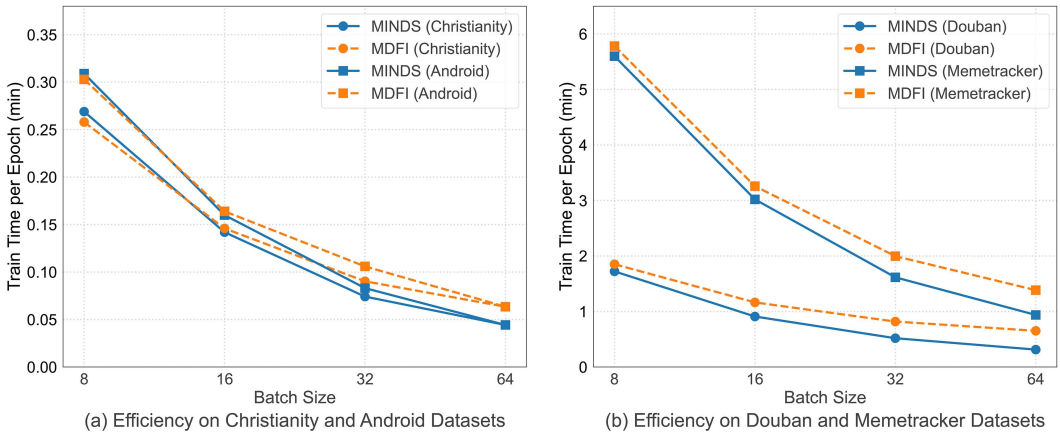

Figure 8 (Color online) Time efficiency analysis on four datasets. (a) Comparison of two models on `Christianity` and `Android` datasets; (b) comparison of two models on `Douban` and `Memetracker` dataests.

also analyze the efficiency of MINDS. From the results illustrated in the Figures 8(a) and (b), we can find that across all tested configurations, the average training time (per epoch) of MDFI remains on the same order of magnitude as that of MINDS. Furthermore, the efficiency curves of both models exhibit a highly similar trend as the batchsize increases, indicating that MDFI has a comparable computational complexity growth pattern with MINDS. Considering the performance improvements that MDFI achieves across four datasets, we think that this modest increase in computational time is entirely acceptable, which strikes a balance between performance and efficiency.

6 Conclusion

In this paper, we propose an MDFI model that simultaneously handles macroscopic and microscopic prediction tasks. Specifically, at the macro-level, we construct a user-cascade graph to model the complex latent interactions between users as well as users and cascades. At the micro-level, we model the social homophily between users. Furthermore, we propose to apply cascade neighbor contrastive learning to constrain user similarity and use a shared representation layer to extract shared features. The experimental results demonstrate the effectiveness of our model.

In future work, we will continue exploring the synergies between the macroscopic and microscopic prediction tasks, and incorporate timestamp information into the user-cascade graph to capture user propagation dynamics. Additionally, another critical direction involves incorporating richer semantic information. Due to the limitations of public benchmark datasets, MDFI lacks access to detailed user profiles or message content text. Thus, we plan to extract and integrate content embeddings (e.g., using pre-trained language models) alongside user attributes (e.g., interest tag) to gain a deeper understanding of the diffusion mechanism in future research. Finally, we acknowledge the inherent scalability challenges when applying our graph construction and contrastive learning mechanism to massive, industry-level networks. Future work will explore optimization strategies such as graph sampling techniques or distributed learning frameworks to enhance the model's efficiency for ultra-large-scale diffusion prediction.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2022YFB3102600), National Natural Science Foundation of China (Grant Nos. U25B2047, U23A20296, 62272469, 72371244, 92467302, 72421002), National Science and Technology Major Project for Brain Science and Brain-like Intelligence Technology (Grant No. 2025ZD0215700), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM202), Major Program of Xiangjiang Laboratory (Grant No. 24XJJCYJ01001), and Science and Technology Innovation Program of Hunan Province (Grant No. 2023RC1007).

References

1 Sun L, Rao Y, Lan Y, et al. HG-SL: jointly learning of global and local user spreading behavior for fake news early detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 5248–5256
2 Leskovec J, Adamic L A, Huberman B A. The dynamics of viral marketing. ACM Trans Web, 2007, 1: 5
3 Yang C, Wu Q, Gao X, et al. EPOC: A survival perspective early pattern detection model for outbreak cascades. In: Proceedings of DEXA, 2018. 336–351
4 Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence through a social network. Theor Comput, 2015, 11: 105–147
5 Wang Y, Shen H, Liu S, et al. Learning user-specific latent influence and susceptibility from information cascades. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2015. 477–484
6 Li C, Ma J, Guo X, et al. Deepcas: An end-to-end predictor of information cascades. In: Proceedings of the International World Wide Web Conferences, 2017. 577–586

7 Wang J, Zheng V W, Liu Z, et al. Topological recurrent neural network for diffusion prediction. In: Proceedings of the IEEE International Conference on Data Mining, 2017. 475–484
8 Yang C, Tang J, Sun M, et al. Multi-scale information diffusion prediction with reinforced recurrent networks. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2019. 4033–4039
9 Chen X, Zhang K, Zhou F, et al. Information cascades modeling via deep multi-task learning. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019. 885–888
10 Jiao P, Chen H, Bao Q, et al. Enhancing multi-scale diffusion prediction via sequential hypergraphs and adversarial learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 8571–8581
11 Wu T, Chen L, Xian X, et al. Evolution prediction of multi-scale information diffusion dynamics. Knowl-Based Syst, 2016, 113: 186–198
12 Wu T, Liu Q, Cao Y, et al. Continual graph convolutional network for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 13754–13762
13 Yuan C, Li J, Zhou W, et al. Dyhgcn: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2020. 347–363
14 Bao P, Shen H, Huang J, et al. Popularity prediction in microblogging network: a case study on Sina Weibo. In: Proceedings of the International World Wide Web Conference (Companion Volume), 2013. 177–178
15 Romero D M, Tan C, Ugander J. On the interplay between social and topical structure. In: Proceedings of the International AAAI Conference on Web and Social Media, 2013
16 Kong S, Mei Q, Feng L, et al. Predicting bursts and popularity of hashtags in real-time. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2014. 927–930
17 Barbieri N, Bonchi F, Manco G. Topic-aware social influence propagation models. Knowl Inf Syst, 2013, 37: 555–584
18 Zhang J, Tang J, Zhong Y, et al. Structinf: Mining structural influence from social streams. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2017. 73–80
19 Mishra S, Rizoiu M, Xie L. Feature driven and point process approaches for popularity prediction. In: Proceedings of the ACM International Conference on Information and Knowledge Management, 2016. 1069–1078
20 Zhao Q, Erdogdu M A, He H Y, et al. SEISMIC: A self-exciting point process model for predicting tweet popularity. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2015. 1513–1522
21 Cao Q, Shen H, Cen K, et al. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In: Proceedings of the ACM International Conference on Information and Knowledge Management, 2017. 1149–1158
22 Cao Q, Shen H, Gao J, et al. Popularity prediction on social platforms with coupled graph neural networks. In: Proceedings of the ACM International Conference on Web Search and Data Mining, 2020. 70–78
23 Chen X, Zhou F, Zhang K, et al. Information diffusion prediction via recurrent cascades convolution. In: Proceedings of the IEEE International Conference on Data Engineering, 2019. 770–781
24 Zhou F, Xu X, Zhang K, et al. Variational information diffusion for probabilistic cascades prediction. In: Proceedings of the IEEE International Conference on Computer Communications, 2020. 1618–1627
25 Zhang Y, Lyu T, Zhang Y. COSINE: community-preserving social network embedding from information diffusion cascades. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018. 2620–2627
26 Bourigault S, Lagnier C, Lamprier S, et al. Learning social network embeddings for predicting information diffusion. In: Proceedings of the ACM International Conference on Web Search and Data Mining, 2014. 393–402
27 Islam M R, Muthiah S, Adhikari B, et al. Deepdiffuse: Predicting the 'who' and 'when' in cascades. In: Proceedings of the IEEE International Conference on Data Mining, 2018. 1055–1060
28 Yang C, Sun M, Liu H, et al. Neural diffusion model for microscopic cascade study. IEEE Trans Knowl Data Eng, 2021, 33: 1128–1139
29 Sankar A, Zhang X, Krishnan A, et al. Inf-vae: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In: Proceedings of the ACM International Conference on Web Search and Data Mining, 2020. 510–518
30 Sun L, Rao Y, Zhang X, et al. MS-HGAT: memory-enhanced sequential hypergraph attention network for information diffusion prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022. 4156–4164
31 Liu B, Yang D, Shi Y, et al. Improving information cascade modeling by social topology and dual role user dependency. In: Proceedings of the International Conference on Database Systems for Advanced Applications, 2022. 425–440
32 Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning, 2020. 1597–1607
33 Yu J, Yin H, Li J, et al. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In: Proceedings of the International World Wide Web Conference, 2021. 413–424
34 Sun T, Qian Z, Dong S, et al. Rumor detection on social media with graph adversarial contrastive learning. In: Proceedings of the International World Wide Web Conference, 2022. 2789–2797
35 Wang Y, Cai Y, Liang Y, et al. Adaptive data augmentation on temporal graphs. In: Proceedings of the Conference on Neural Information Processing Systems, 2021. 1440–1452
36 Bielak P, Kajdanowicz T, Chawla N V. Graph barlow twins: A self-supervised representation learning framework for graphs. Knowl-Based Syst, 2022, 256: 109631
37 Gong X, Yang C, Shi C. MA-GCL: model augmentation tricks for graph contrastive learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 4284–4292
38 Shen X, Sun D, Pan S, et al. Neighbor contrastive learning on learnable graph augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 9782–9791
39 Cheng Z, Ye W, Liu L, et al. Enhancing information diffusion prediction with self-supervised disentangled user and cascade representations. In: Proceedings of the ACM International Conference on Information and Knowledge Management, 2023. 3808–3812
40 Yao L, Mao C, Luo Y. Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 7370–7377
41 Zhong E, Fan W, Wang J, et al. Comsoc: adaptive transfer of user behaviors over composite social network. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2012. 696–704
42 Leskovec J, Backstrom L, Kleinberg J M. Meme-tracking and the dynamics of the news cycle. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2009. 497–506
43 Wu D, Tan Z, Xia Z, et al. TCSE: Trend and cascade based spatiotemporal evolution network to predict online content popularity. Multimed Tools Appl, 2023, 82: 1459–1475
44 Xu X, Zhou F, Zhang K, et al. CasFlow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. IEEE Trans Knowl Data Eng, 2023, 35: 3484–3499