Enhancing Multi-Scale Diffusion Prediction via Sequential Hypergraphs and Adversarial Learning

Pengfei Jiao[1, 4], **Hongqian Chen**[1], **Qing Bao**[1], **Wang Zhang**[2], **Huaming Wu**[3*]

[1]School of Cyberspace, Hangzhou Dianzi University, China
[2]College of Intelligence and Computing, Tianjin University, China
[3]Center for Applied Mathematics, Tianjin University, China
[4]Data Security Governance Zhejiang Engineering Research Center, Hangzhou Dianzi University, China
{pjiao, hqchen, qbao}@hdu.edu.cn, {wangzhang, whming}@tju.edu.cn

Abstract

Information diffusion prediction plays a crucial role in understanding the propagation of information in social networks, encompassing both macroscopic and microscopic prediction tasks. Macroscopic prediction estimates the overall impact of information diffusion, while microscopic prediction focuses on identifying the next user to be influenced. While prior research often concentrates on one of these aspects, a few tackle both concurrently. These two tasks provide complementary insights into the diffusion process at different levels, revealing common traits and unique attributes. The exploration of leveraging common features across these tasks to enhance information prediction remains an underexplored avenue. In this paper, we propose an intuitive and effective model that addresses both macroscopic and microscopic prediction tasks. Our approach considers the interactions and dynamics among cascades at the macro level and incorporates the social homophily of users in social networks at the micro level. Additionally, we introduce adversarial training and orthogonality constraints to ensure the integrity of shared features. Experimental results on four datasets demonstrate that our model significantly outperforms state-of-the-art methods.

Introduction

Online social platforms have become an essential part of our daily lives, enriching instant communication among individuals and expediting the swift dissemination of information. The activity patterns of users in social networks play a pivotal role in the spread of information, leading to the emergence of information cascades. Gaining a deeper understanding of the underlying mechanisms of information diffusion carries significant economic and social advantages, with applications in various fields, including fake news detection (Zhang et al. 2023), viral marketing (Miller and Lammas 2010), and recommender system (Ko et al. 2022).

As shown in Fig. 1, current researches on modeling information cascades primarily focus on two key aspects: 1) **Macroscopic prediction**, such as DeepCas (Li et al. 2016) and CasCN (Chen et al. 2019b), estimating the incremental or total size of a cascade; 2) **Microscopic prediction**, such as TopoLSTM (Wang et al. 2017) and SNIDSA (Wang, Chen, and Li 2018), predicting the subsequent user to be influenced within the cascade.

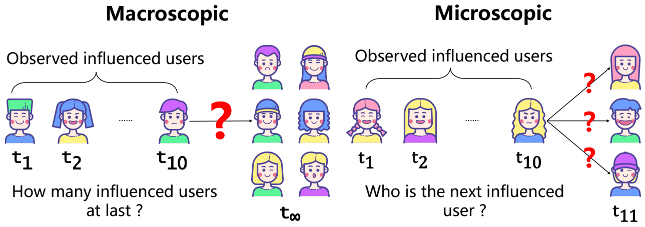

Macroscopic
Microscopic
Observed influenced users
Observed influenced users
t_1
t_2
t_{10}
t_1
t_2
t_{10}
?
?
?
?
How many influenced users at last ?
t_∞
Who is the next influenced user ?
t_{11}

Figure 1: Illustrations depicting macroscopic cascade size prediction (left) and microscopic next influenced user prediction (right).

On the one hand, macro-prediction concentrates on overarching patterns and trends, employing network topology and dissemination models to forecast information propagation. On the other hand, micro-prediction delves into the particulars of individual users' behaviors and attributes, utilizing analyses of user and content characteristics to anticipate the impact of information diffusion. Macro-prediction and micro-prediction collectively provide a comprehensive understanding of information dissemination across various levels and can mutually reinforce and enhance each other. Since both tasks require learning propagation features from observed cascades, they inherently share commonalities. Hence, the imperative to enhance prediction accuracy by extracting common features between these tasks assumes paramount importance.

However, the extraction of such common features is confronted with challenges. Firstly, information dissemination involves complex interactions not only within a given cascade but also between different cascades. Moreover, the evolution of cascades over time demands an approach capable of encapsulating both global interactions and dynamic changes. Secondly, ensuring the purity of public features in the presence of potential contamination by private features poses a significant challenge.

To the best of our knowledge, only a limited number of studies have introduced a unified model catering to both macro and micro scales. The most representative works are FOREST (Yang et al. 2019) and DMT-LIC (Chen et al. 2019a). Nevertheless, FOREST (Yang et al. 2019) primarily

[*]Corresponding author

utilizes the outcomes of micro-prediction to guide macro-prediction, lacking a comprehensive recognition of the mutually reinforcing synergy inherent in these two tasks. Similarly, while DMT-LIC (Chen et al. 2019a) incorporates a shared representation layer to capture cascade graph representations and diffusion processes, it fails to address the issue of potential contamination and redundancy between the shared features and task-specific features. Moreover, both methods primarily concentrate on user interactions within individual cascades, neglecting the intricate interactions and dynamics among cascades at a global level.

To address the above challenges, we propose MINDS, a streamlined and efficient model for **M**ulti-scale **IN**formation **D**iffu**S**ion prediction. Specifically, at the macro level, we construct sequential hypergraphs to effectively capture the interactions and dynamics among cascades. From a global perspective, modeling complex interactions among users and cascades is consistent with the concept of the hypergraph. Constructing sequential hypergraphs by dividing the time period into sequential time windows can accurately describe the dynamic evolution of the cascades. At the micro level, we focus on understanding the social homophily among users within social networks. We design a shared module to learn shared features for both macro and micro tasks. Furthermore, we incorporate adversarial training and orthogonality constraints to mitigate feature redundancy and contamination between shared and task-specific features. In summary, the main contributions of this paper are three-fold:

- We propose an effective and straightforward model that tackles both macro and micro prediction, leveraging their mutual reinforcement to enhance overall performance.
- We introduce an approach that captures the interactions and dynamics among cascades by modeling information diffusion in sequential hypergraphs. To address feature redundancy, we incorporate adversarial training and orthogonality constraints.
- We conducted comprehensive experiments to evaluate our model's performance. The results demonstrate its superiority over state-of-the-art methods in both macro and micro prediction.

Problem Formulation

To commence, we present the social graph and diffusion hypergraphs that constitute the foundation for diffusion prediction within our model. The social graph is denoted as $G_S = (U, E)$, where U is the user set and E is the edge set. Each edge $(u_i, u_j) \in E$ represents a social relationship between user u_i and u_j. The observed diffusion cascades $D = \{d_1, d_2, \ldots, d_M\}, |D| = N$ are split into T subsets according to timestamps for constructing sequential diffusion hypergraphs $G_D = \{G_D^t | t = 1, 2, \ldots, T\}, G_D^t = (U^t, \mathcal{E}^t)$, where U^t is the user set and $\mathcal{E}^t$ is the hyperedge set. In the diffusion hypergraph, users participate in the same cascade and are connected by a hyperedge, in other words, a hyperedge represents a cascade. Note that the set of nodes connected by hyperedge is different in each hypergraph. It means that if u_i participates in d_m during the t-th time interval, then u_i being connected to hyperedge e_m only occurs in diffusion hypergraph G_D^t.

In this work, we aim to address both the macroscopic and microscopic diffusion prediction problems based on the above introductions.

Macroscopic Diffusion Prediction: Given a social graph G_S, diffusion hypergraphs G_D and an observed diffusion sequence $d_m = \{(u_i^m, t_i^m) | u_i^m \in U\}$, estimate the final size $|d_m|$ of cascade d_m.

Microscopic Diffusion Prediction: Given a social graph G_S, diffusion hypergraphs G_D and an observed diffusion sequence $d_m = \{(u_i^m, t_i^m) | u_i^m \in U\}$, predict which user will participate in d_m in the next step.

Method

In this section, we will provide a comprehensive introduction to the proposed model. The architectural overview of the proposed model is depicted in Fig. 2, which comprises four primary modules:

User Global Interactive Learning Module: This module is responsible for extracting user preferences at each time interval and characterizing the dynamic changes of cascades. A fusion layer at the cascade level facilitates this process.

User Social Homophily Learning Module: It captures users' social relationship at the individual user level using Graph Convolutional Networks (GCN).

Shared-private Representation Learning Module: This module learns task-specific representations and shared representations to facilitate diffusion prediction.

Diffusion Prediction Module: This module concatenates task-specific features with shared representation for macroscopic and microscopic diffusion prediction, respectively.

User Global Interactive Learning

In order to simultaneously account for global interactions among cascades and dynamic changes of cascades. On the basis of the constructed sequential diffusion hypergraphs, we introduce the HGNN to learn the user global interactions of each independent time interval at the cascade level, and add a fusion layer between two continuous time intervals to model the dynamics of cascades.

Hypergraph Neural Network At each time interval, we model the interactions of users through HGNN. The process of HGNN is illustrated in Fig. 3. For a simple graph, graph convolution takes the aggregation of its neighbor vertices to get a new representation of the central vertex. The information of vertices is passed through edges in a simple graph. Similarly, hyperedges play a role in information transmission in a hypergraph. The message aggregating in the hypergraph can be summarized in a two-stage procedure: 1) Vertex-to-Hyperedge; 2) Hyperedge-to-Vertex.

Vertex-to-Hyperedge. Given a diffusion hypergraph G_D^t, the first stage of HGNN aims to update the feature $y_{j,t}$ of hyperedge e_j^t by aggregating the information of all its connected vertices, which can be defined as:

$$y_{j,t}^l = \sigma\left(w_{e_j^t} \cdot \sum_{u_i^t \in \mathcal{N}_v(e_j^t)} \frac{x_{i,t}^l}{|\mathcal{N}_v(e_j^t)|}\right), \tag{1}$$

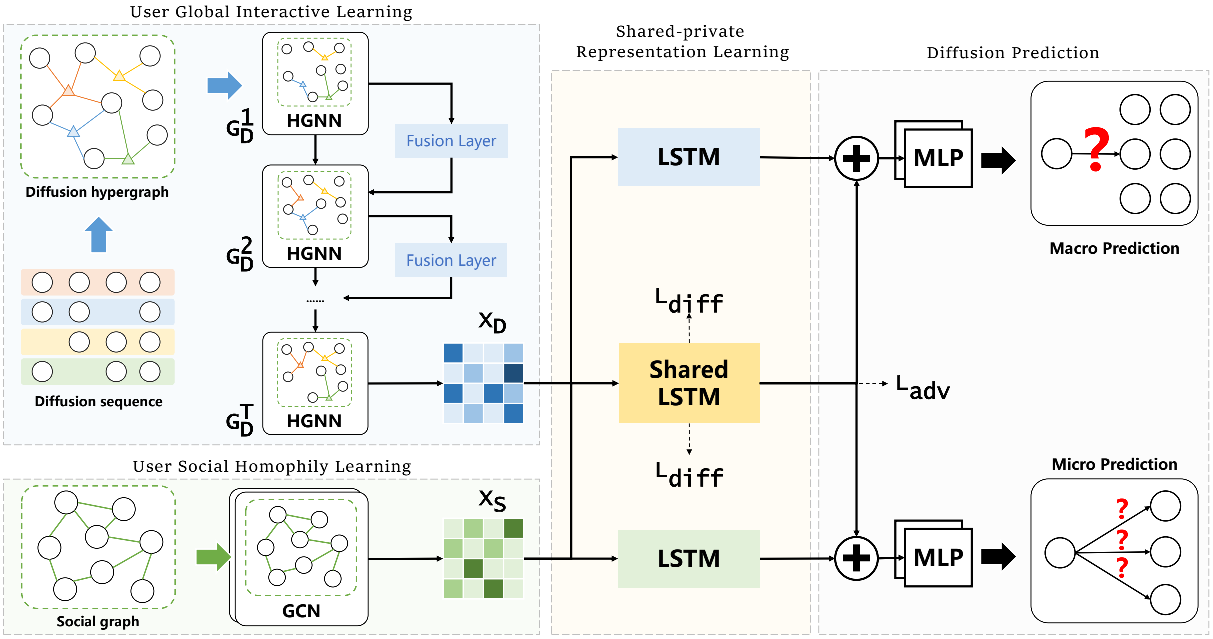


Figure 2: The architectural overview of our model.

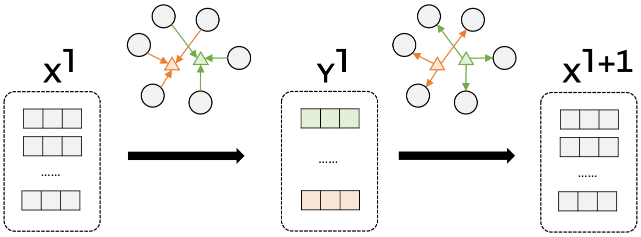


Figure 3: The two stages of hypergraph convolution.

where σ is an non-linear activation function $\mathbf{ReLU}$ and $\mathcal{N}_v(e_j^t)$ is the set of vertices connected by hyperedge e_j^t. $w_{e_j^t}$ is a weight associated to hyperedge e_j^t. We consider each cascade to be of equal importance and give the same weight to each hyperedge when aggregating, i.e. $w_{e_j^t} = 1$.

Hyperedge-to-Vertex. After updating features of hyperedges, the second stage aims to aggregate the information of all hyperedges participated by u_i^t for updating the feature $x_{i,t}$ of u_i^t at t-th time interval. The update process can be defined as:

$$x_{i,t}^{l+1} = \sigma\left(\Theta^l \cdot \sum_{e_j^t \in \mathcal{N}_e(u_i^t)} \frac{y_{j,t}^l}{|\mathcal{N}_e(u_i^t)|}\right), \tag{2}$$

where $\mathcal{N}_e(u_i^t)$ is the set of hyperedges connected by vertex u_i^t. $\Theta^l \in \mathbb{R}^{d \times d}$ is a trainable parameter of layer l and d is the dimension of embedding.

Sequential HGNNs with Fusion Layer The above two-stage convolution operation only learns user interaction at a specific time interval, which can not adequately characterize the evolution of cascades in propagation. Therefore, we design a fusion strategy to connect the interactions at different time intervals learned by HGNN in chronological order. The fusion strategy is defined as:

$$\begin{aligned} x_{i,t+1}^0 &= \alpha x_{i,t}^L + (1-\alpha) x_{i,t}^0 \\ \alpha &= \frac{\exp(\mathbf{W}_{F_2}^T \sigma(\mathbf{W}_{F_1} x_{i,t}^L))}{\exp(\mathbf{W}_{F_2}^T \sigma(\mathbf{W}_{F_1} x_{i,t}^L)) + \exp(\mathbf{W}_{F_2}^T \sigma(\mathbf{W}_{F_1} x_{i,t}^0))}, \end{aligned} \tag{3}$$

where $x_{i,t}^0$ is the initial feature of user u_i^t and $x_{i,t}^L$ is the updated feature of user u_i^t learned from diffusion hypergraph G_D^t through L-layer HGNN. At the first time interval, we initialize the user feature embedding from a normal distribution. $\sigma(\cdot)$ is the activation function $\mathbf{ReLU}$. $\mathbf{W}_{F_1}$ represents the transformation matrix, while $\mathbf{W}_{F_2}^T$ denotes the vector used for calculating attention scores.

We can obtain the final global interactive representation $\mathbf{X}_D$ through sequential HGNNs.

User Social Homophily Learning

User tends to have more social interactions with users who are similar to them and this refers to the principle called *social homophily*. Close friends, who are usually friends alike in certain qualities or interests, have more influence on each other than dissimilar ones. Users' social homophily can be reflected through social network structure. We introduce the social graph to model user social relationships and apply a multi-layer GCN to embed social homophily. Given social graph $G_S = (U, E)$, the user social homophily embedding matrix $\mathbf{X}_S^l$ at l-th layer is updated by:

$$\mathbf{X}_S^{l+1} = \sigma(\tilde{\mathbf{D}}_S^{-\frac{1}{2}} \tilde{\mathbf{A}}_S \tilde{\mathbf{D}}_S^{-\frac{1}{2}} \mathbf{X}_S^l \mathbf{W}_S), \tag{4}$$

where σ is the $\mathbf{ReLU}$ activation function, $\mathbf{W}_S$ is a trainable weight matrix, $\tilde{\mathbf{A}}_S$ and $\tilde{\mathbf{D}}_S$ are the adjacent and degree matrix of self-looped G_S. The initial homophily embedding matrix $\mathbf{X}_S^0 \in \mathbb{R}^{N\times d}$ is randomly initialized from a normal distribution, and d is the dimension of embedding. We can obtain the final social homophily representation $\mathbf{X}_S$ after several layers of GCN.

Shared-private Representation Learning

Graph-based representation learning captures the co-occurrence relationship of users at the user level and cascade level, however, it does not enable further analysis of context interactions within cascades. Therefore, due to the excellent performance of the LSTM in sequential tasks such as natural language processing, we apply two LSTM modules to learn the social and global context interactions within cascades, respectively.

We hold the belief that there exists a hidden common feature between the macro-prediction and the micro-prediction tasks, with the potential to enhance the performance of each task individually. Drawing inspiration from the principles of multi-task learning (Liu, Qiu, and Huang 2017), we propose a shared LSTM architecture aimed at capturing these shared characteristics between the two tasks. Furthermore, to tackle the issue of feature redundancy, we introduce a combination of adversarial training and orthogonality constraints.

Private Representation Learning We utilize the LSTM to model the cascade diffusion process sequentially, where a hidden state is employed to capture the diffusion history.

The update of each LSTM unit can be shortened as:

$$h_t = \mathbf{LSTM}(h_{t-1}, x_t, \theta_p), \tag{5}$$

where $h_t \in \mathbb{R}^d$ is the hidden state and $x_t \in \mathbb{R}^d$ is the input at the current time step. θ_p represents all the parameters in LSTM.

Based on the defined LSTM, we can compute the representation of the context interaction for the user social homophily $\mathbf{X}_S$ and user global interaction matrix $\mathbf{X}_D$ as follows:

$$\begin{aligned} h_t^{cas} &= \mathbf{LSTM}(h_{t-1}^{cas}, x_t^d, \theta_{cas}) \\ h_t^{user} &= \mathbf{LSTM}(h_{t-1}^{user}, x_t^s, \theta_{user}), \end{aligned} \tag{6}$$

where $\mathbf{LSTM}(.,\theta)$ is defined as Eq. 5.

Thus, the task-specific embeddings are represented as $H^{cas} \in \mathbb{R}^{N\times d}$ and $H^{user} \in \mathbb{R}^{N\times d}$.

Shared Representation Learning Inspired by the gated mechanisms used in LSTM, we design a novel shared-LSTM, that takes $\mathbf{X}_D$ and $\mathbf{X}_S$ as the input. The detail of the module is described as follows:

$$\begin{aligned} f_t &= \sigma(x_{t-1}^D W_f + x_{t-1}^S U_f + h_{t-1} V_f + b_f), \\ i_t &= \sigma(x_{t-1}^D W_i + x_{t-1}^S U_i + h_{t-1} V_i + b_i), \\ o_t &= \sigma(x_{t-1}^D W_o + x_{t-1}^S U_o + h_{t-1} V_o + b_o), \\ \tilde{c}_t &= \tanh(x_{t-1}^D W_c + x_{t-1}^S U_c + h_{t-1} V_c + b_c), \\ c_t &= \tilde{c}_t \cdot i_t + c_{t-1} \cdot f_t, \qquad h_t = o_t \cdot \tanh c_t, \end{aligned} \tag{7}$$

where σ is the sigmoid function. $W_* \in \mathbb{R}^{d\times d}$, $U_* \in \mathbb{R}^{d\times d}$, $V_* \in \mathbb{R}^{d\times d}$ and $b_* \in \mathbb{R}^d$ are trainable parameters. The input gate i_t controls the amount of new information added to the hidden state, while the forget gate f_t regulates the amount of information discarded from the previous memory cells c_t. Additionally, the output gate o_t determines the amount of information to be output in the hidden state h_t. By integrating the forget gate, input gate, update memory unit, and output gate, the shared LSTM can effectively handle the intricate relationship between micro-features and macro-features.

We finally obtain a comprehensive representation of shared features between the macro-prediction task and micro-prediction task from the shared LSTM, which is denoted as $H^{share} \in \mathbb{R}^{N\times d}$.

Adversarial Training Although the shared-private LSTM is designed to learn the shared and task-specific features, there is no guarantee that shared features can not be preserved in private feature space, or vice versa. Therefore, a simple principle can be applied to shared LSTM that a reliable shared feature should primarily consist of common information without any task-specific information. Inspired by adversarial networks, we introduce adversarial training to solve this problem.

A task discriminator is used to map the representation into a probability distribution, estimating which tasks the encoded feature comes from.

$$D(h, \theta_D) = softmax(b + Uh), \tag{8}$$

where $U \in \mathbb{R}^{d\times d}$ is a learnable parameter and $b \in \mathbb{R}^d$ is a bias.

To prevent task-specific features from infiltrating the shared representation, we design a task adversarial loss, denoted as L_{adv}. This loss function is employed to train the model in such a way that the shared features generated are not easily predictable by a classifier in terms of their corresponding tasks. Formally, the task adversarial loss, L_{adv}, is defined as follows:

$$L_{adv} = \min_{\theta_{share}} \max_{\theta_D} \sum_{k=1}^{2} \sum_{n=1}^{N} (\log D(h_n^k) + \log(1 - D(h_n^{share}))), \tag{9}$$

where θ_{share} represents all the parameters in Shared-LSTM, and k denotes the task type (either macro or micro). The optimization process involves a min-max framework, with the underlying concept being that the shared LSTM generates a representation to intentionally confuse the task discriminator. As the training progresses, the shared feature extractor and task discriminator gradually reach a point of convergence, beyond which achieving additional enhancements becomes challenging. As a result, the task discriminator becomes progressively incapable of distinguishing among various tasks. This convergence indicates the successful acquisition of shared feature generation by the feature extractor, resulting in shared features that exhibit indistinguishability across all tasks.

Orthogonality Constraints It is worth noting that the above model has a potential disadvantage. The disadvantage is that task-invariant features can appear in both shared and private representations. To alleviate this drawback, we introduce orthogonality constraints, which penalize redundant latent representations and encourage the shared and private

LSTM to encode different aspects of the inputs. The orthogonality constraints are defined as:

$$L_{diff} = \left\| {H^{share}}^T H^{cas} \right\|_F^2 + \left\| {H^{share}}^T H^{user} \right\|_F^2, \quad (10)$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm.

Diffusion Prediction

We concatenate the task-specific representation H^{cas} and H^{user} for each task with the shared representation H^{share}, respectively. These concatenated representations are then separately fed into distinct output layers dedicated to the prediction process.

Macroscopic Diffusion Prediction For macroscopic diffusion prediction, we aim to predict the final cascade size in the future. We calculate the final size of diffusion cascade d_m by:

$$S_m = \mathrm{MLP}(\mathrm{concat}(h^{cas}, h^{share})), \quad (11)$$

where $\mathrm{concat}(\cdot,\cdot)$ is the concatenation operation.

We train the macroscopic task by minimizing the following loss function:

$$L_{macro} = \frac{1}{M}\sum_{m=1}^{M}(S_m - \hat{S}_m)^2, \quad (12)$$

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

Microscopic Diffusion Prediction For microscopic diffusion prediction, we predict the next influenced probability $p_i \in \mathbb{R}^{|d_m|}$ for user u_i:

$$p_i = \mathrm{softmax}(\mathrm{MLP}(\mathrm{concat}(h^{user}, h^{share}))). \quad (13)$$

We adopt the cross entropy loss for microscopic training:

$$L_{micro} = -\sum_{j=2}^{|d_m|}\sum_{i=1}^{|U|} \hat{p}_{ji}\log(p_{ji}), \quad (14)$$

where $|U|$ is the number of users and $\tilde{p}$ is true probability. If user u_i participate in cascade d_m at the step j, then $\hat{p}_{ji} = 1$, otherwise $\hat{p}_{ji} = 0$.

The overall loss function of our model is defined as:

$$L = \lambda L_{macro} + (1-\lambda)L_{micro} + L_{adv} + \gamma L_{diff}, \quad (15)$$

where λ is a balance parameter and γ is a hyperparameter.

Experiment

In this section, we conduct experiments on both microscopic and macroscopic cascade predictions to demonstrate the effectiveness of our proposed model.

Experimental Setting

Datasets We conduct experiments on four datasets, i.e., Christianity, Android, Douban and Memetracker. The statistics of these datasets are shown in Table 1. A detailed description of the datasets can be found in the Appendix.

Dataset	Christ	Android	Douban	Meme
# Users	2,897	9,958	12,232	4,709
# Links	35,624	48,573	39,658	209,194
# Cascades	589	679	3,475	12,661
Avg. Length	22.9	33.3	21.76	16.24

Table 1: Statistics of datasets. Christ is short for the dataset Christianity, and Meme is short for the dataset Memetracker.

Baselines We compare thirteen representative baseline models with our models.

For macroscopic prediction, we evaluate five models: DeepCas (Li et al. 2016), DeepHawkes (Cao et al. 2017), CasCN (Chen et al. 2019b), CasFlow (Xu et al. 2023b) and TCSE-net (Wu et al. 2022).

For microscopic prediction, we evaluate six models: TopoLSTM (Wang et al. 2017), NDM (Yang et al. 2021), SNIDSA (Wang, Chen, and Li 2018), Inf-VAE (Sankar et al. 2020), DyHGCN (Yuan et al. 2020) and TAN-DRUD (Liu et al. 2022).

For multi-scale prediction, we evaluate two models: FOREST (Yang et al. 2019) and DMT-LIC (Chen et al. 2019a).

A detailed description of baselines can be found in the Appendix.

Evaluate Metrics For macroscopic prediction, we use *Mean Squared Logarithmic Error* (MSLE) as the evaluation metric, which is also used in previous experiments (Cao et al. 2017; Li et al. 2016). For microscopic prediction, we use two ranking metrics used in (Yang et al. 2019): *Mean Average Precision* on top *k* (MAP@*k*) and *Hits Scores* on top *k* (Hits@*k*) for evaluation, $k = [10, 50, 100]$.

Parameters Settings For each dataset, we employ a random sampling method to allocate 80% of cascades for training, 10% for validation, and the remaining 10% for testing. Baseline methods follow the original paper settings. For MINDS, we implement the model using PyTorch and utilize the Adam optimizer with a learning rate of 0.001. The embedding dimension is set to 64, and the batch size is 32. The balance parameter λ is assigned a value of 0.3, while the hyperparameter γ is set to 0.05. Social homophily learning utilizes a 2-layer GCN, and global interaction learning is facilitated through a single-layer HGNN. Additionally, the number of time intervals is set to 8.

Performance Comparison

We conduct a comprehensive comparison of MINDS with various baselines on four datasets, focusing on microscopic and macroscopic diffusion prediction. The results are summarized in Tables 2, 3, and 4, and we observe the following:

1) MINDS consistently outperforms all state-of-the-art baselines in microscopic prediction tasks. Compared to the second-best model DyHGCN, MINDS leverages sequential hypergraphs to dynamically represent cascade interactions, leading to remarkable improvements of up to 3% in Hits scores and MAP scores.

2) MINDS consistently outperforms all state-of-the-art baselines in macroscopic prediction tasks, achieving at least

Models	**Christianity**			**Android**			**Douban**			**Memetracker**		
	@10	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
TopoLSTM	0.1559	0.3653	0.4777	0.0460	0.1318	0.2103	0.0306	0.0143	0.0184	0.1908	0.3687	0.4683
NDM	0.0464	0.1145	0.1461	0.0170	0.0423	0.0555	0.0388	0.0506	0.0528	0.0931	0.1228	0.1279
SNIDSA	0.0660	0.2098	0.3502	0.0271	0.0829	0.1299	0.0702	0.1807	0.2324	0.1395	0.2945	0.3977
Inf-VAE	0.0767	0.2569	0.3853	0.0318	0.0938	0.1452	0.1364	0.2361	0.3059	0.1165	0.3096	0.4200
DyHGCN	0.2380	0.4689	0.5923	0.0748	0.1746	0.2596	0.1438	0.2648	0.3329	0.2522	0.4603	0.5710
TAN-DURD	0.1908	0.4406	0.5697	0.0281	0.1024	0.1658	0.0841	0.1604	0.2175	0.2139	0.4247	0.5383
FOREST	0.2746	0.4665	0.5603	0.0866	0.1739	0.2314	0.1106	0.1986	0.2559	0.2648	0.4502	0.5499
DMT-LIC	0.2768	0.4442	0.5669	0.0932	0.1639	0.2315	0.1465	0.2506	0.3054	0.2746	0.4619	0.5656
MINDS	**0.3214**	**0.4978**	**0.6250**	**0.1096**	**0.1989**	**0.2766**	**0.1956**	**0.3087**	**0.3641**	**0.2819**	**0.4760**	**0.5790**

Table 2: Results on four datasets (*Hits@k* scores for k = 10, 50 and 100), where higher scores indicate better performance.

Models	**Christianity**			**Android**			**Douban**			**Memetracker**		
	@10	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
TopoLSTM	0.0523	0.0619	0.0635	0.0166	0.0202	0.0213	0.0354	0.0824	0.0884	0.0870	0.0955	0.0969
NDM	0.0144	0.0177	0.0182	0.0059	0.0070	0.0072	0.0141	0.0824	0.0884	0.0463	0.0480	0.0481
SNIDSA	0.0246	0.0306	0.0326	0.0100	0.0122	0.0129	0.0371	0.0419	0.0148	0.0605	0.0674	0.0689
Inf-VAE	0.0172	0.0254	0.0272	0.0076	0.0103	0.0110	0.0543	0.0588	0.0598	0.0425	0.0509	0.0525
DyHGCN	0.1062	0.1167	0.1184	0.0392	0.0434	0.0446	0.0801	0.0856	0.0865	0.1410	0.1502	0.1518
TAN-DURD	0.0752	0.1167	0.1184	0.0099	0.0130	0.0139	0.0359	0.0401	0.0409	0.0991	0.1086	0.1102
FOREST	0.1569	0.1658	0.1672	0.0628	0.0667	0.0675	0.0655	0.0694	0.0702	0.1429	0.1514	0.1528
DMT-LIC	0.1649	0.1728	0.1746	0.0622	0.0652	0.0662	0.0812	0.0856	0.0897	0.1496	0.1581	0.1595
MINDS	**0.1955**	**0.2037**	**0.2054**	**0.0677**	**0.0716**	**0.0727**	**0.1142**	**0.1199**	**0.1213**	**0.1535**	**0.1623**	**0.1638**

Table 3: Results on four datasets (*MAP@k* scores for k = 10, 50 and 100), where higher scores indicate better performance.

Model	**Christ**	**Android**	**Douban**	**Meme**
DeepCas	1.446	2.122	2.122	2.231
DeepHawkes	1.111	1.971	1.725	1.143
CasCN	1.046	0.981	1.476	0.967
CasFlow	0.765	1.041	0.465	0.535
TCSE-net	2.391	2.882	1.033	2.285
FOREST	1.726	0.556	0.825	0.621
DMT-LIC	1.692	0.201	0.741	0.701
MINDS	**0.572**	**0.151**	**0.404**	**0.506**

Table 4: Experimental results on four datasets in terms of *MSLE*, where lower scores indicate better performance. Christ is short for the dataset Christianity, and Meme is short for the dataset Memetracker.

a 10% decrease in MSLE. By combining macroscopic and microscopic prediction, MINDS achieves more promising performance.

3) MINDS convincingly outperforms representative baselines on multi-scale prediction tasks. The improvements stem from the pure shared features, avoiding impurities. MINDS' ability to handle both prediction tasks in a single model enables multi-scale information diffusion prediction.

Ablation Study

We conduct ablation studies on the Christianity and Douban datasets to evaluate the individual contributions of different submodules in MINDS.

As shown in Table 5, MINDS achieves the best results compared to other variants, indicating the effectiveness of its design. Specifically, the observations are as follows:

1) Model performance declines after removing L_{adv}, L_{diff}, or both, validating the importance of introducing adversarial training and orthogonality constraints to address feature redundancy.

2) Introducing a series of interactive hypergraphs effectively captures cascade interactions from a global perspective, as demonstrated by the results of *w/o HGNN*.

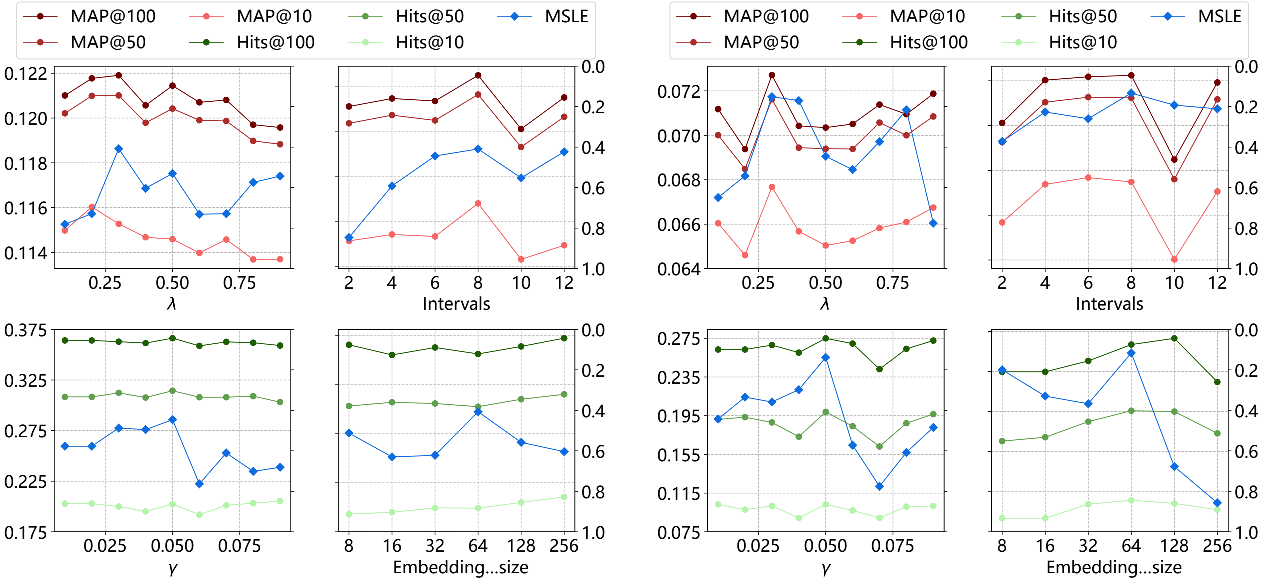
3) Macroscopic prediction improves microscopic prediction by accurately predicting the propagation behavior of individual users. Conversely, microscopic prediction enhances the understanding and interpretation of overall propagation trends by macroscopic prediction. Significant differences between *w/o Macro*, *w/o Micro*, and MINDS in macro and micro indicators reveal the mutual reinforcement between the two tasks, leading to improved performance.

Parameter Analysis

In this subsection, we investigate how different hyperparameter settings affect the performance of our model on the Android and Douban datasets. We explore the sensitivity of λ,

Models	**Christianity**			**Douban**		
	Hits@100	**MAP@100**	**MSLE**	**Hits@100**	**MAP@100**	**MSLE**
w/o AdvDiff	0.5893	0.1958	0.971	0.3682	0.1170	0.642
w/o Diff	0.6004	0.1949	1.222	0.3688	0.1173	0.712
w/o Adv	0.5915	0.1926	0.861	0.3572	0.1193	0.742
w/o HGNN	0.5871	0.2013	1.074	0.3692	0.1178	0.581
w/o Macro	0.5580	0.1874	9.255	0.3665	0.1191	4.669
w/o Micro	0.5871	0.1937	0.865	0.3591	0.1174	0.711
MINDS	**0.6250**	**0.2054**	**0.572**	**0.3736**	**0.1213**	**0.549**

Table 5: Ablation study on Christianity and Douban datasets. We design six variants to demonstrate the rationale behind our model: *w/o AdvDiff* removes L_{adv} and L_{diff}. *w/o Diff* removes L_{diff}. *w/o Adv* removes L_{adv}. *w/o HGNN* replaces sequential hypergraphs with sequential digraphs and HGNN with GAT. *w/o Macro* removes L_{macro}. *w/o Micro* removes L_{micro}.



(a) Douban

(b) Android

Figure 4: Parameter sensitivity on Douban and Android dataset. For balance parameter $\lambda \in (0, 1)$ and the number of time intervals $\in [2, 12]$, we evaluate all map and MSLE scores. For hyper parameter $\gamma \in (0, 0.1)$ and embedding size $\in \{8, 16, 32, 64, 128, 256\}$, we evaluate all hits scores and MSLE score. In this figure, the macro indicator (MSLE) is presented with an inverted Y-axis to align with the increasing trend of the micro indicator (MAP and Hits).

γ, embedding size, and the number of time intervals, testing each parameter while keeping others fixed. Fig. 4 illustrates the model's performance on multi-scale prediction under various hyperparameter configurations.

During the process of parameter value selection, we carefully consider both macro and micro indicators. Optimal model performance occurs when the macro index is minimized, and the micro index is maximized. Remarkably, we observe that MINDS maintains stable performance when hyperparameters are varied within a reasonable range. This experiment highlights the robustness of our model. Finally, we determine that the optimal hyperparameter configuration corresponds to (λ, γ, embedding size, number of time intervals) = (0.3, 0.05, 64, 8).

Conclusion

In this paper, we propose MINDS, a streamlined yet effective multi-scale diffusion prediction model, capable of handling both microscopic and macroscopic predictions. Our approach involves constructing sequential hypergraphs to capture intricate influences and dynamics among cascades from a macro perspective. Simultaneously, we learn implicit structures and user characteristics in social networks from a micro perspective. A shared LSTM is then employed to extract common features between macro- and micro-tasks, while adversarial training and orthogonality constraints ensure the purity of these shared features. Experimental results on the next-influenced user and cascade size predictions demonstrate the effectiveness of our method.

Appendix

Related Work

Macroscopic Diffusion Prediction Previous studies can be categorized into three main approaches: feature-based, generative process-based, and deep learning-based methods.

Feature-based (Kong et al. 2014) approaches focus on extracting handcrafted features from the input data, which are then used in machine learning algorithms for regression or classification tasks. However, these methods heavily rely on domain knowledge and lack generalizability.

Generative process-based approaches (Zhao et al. 2015) model the arrival of infected users as a point process. While these methods enhance interpretability, they may overlook implicit information within the cascade dynamics.

Recently, deep learning-based approaches have shown their effectiveness. For example, DeepCas (Li et al. 2016) utilizes RNN to encode sampled sequences from social graphs and cascades. DeepHawkes (Cao et al. 2017) incorporates the Hawkes process within an RNN architecture. CoupledGNN (Cao et al. 2019) and CasCN (Chen et al. 2019b) utilize GNNs to capture diffusion patterns across the underlying social network. VaCas (Zhou et al. 2020) combines graph wavelets, hierarchical variational autoencoders, and Bi-GRUs to learn the structures of cascade graphs.

Microscopic Diffusion Prediction Conventional methods for microscopic diffusion prediction can be categorized into three groups: independent cascade (IC)-model-based approaches, embedding-based approaches, and deep learning-based approaches.

IC-model-based approaches (Wang et al. 2014) assume independent diffusion probabilities for user pairs and employ Monte Carlo simulations to predict microscopic diffusion.

Embedding-based approaches (Feng et al. 2018a) extend the IC model by representing each user as a parameterized vector. They model diffusion probabilities between users based on their embeddings, considering factors such as global user similarity. However, these methods overlook infection history.

Deep learning techniques have shown promise in modeling information diffusion. Approaches like TopoLSTM (Wang et al. 2017) structure hidden states as directed acyclic graphs, while DeepDiffuse (Islam et al. 2018) and HiDAN (Wang and Li 2019) incorporate attention mechanisms to leverage infection timestamp information. NDM (Yang et al. 2021) combines self-attention and CNNs, while Inf-VAE (Sankar et al. 2020) integrates a VAE framework to capture social homophily and temporal influence. SNIDSA (Wang, Chen, and Li 2018) and DyHGCN (Yuan et al. 2020) utilize diffusion paths, social networks, and temporal information for prediction. Furthermore, methods like MS-HGAT (Sun et al. 2022) and HyperINF (Jin et al. 2022) leverage hypergraphs to learn global user dependencies.

Hypergraph Neural Network Hypergraph offers a natural way to represent group relations by connecting entities through hyperedges. Recently, several approaches have emerged to leverage hypergraphs for learning latent node representations and capturing high-order structural information.

HGNN (Feng et al. 2018b) stands as the pioneering spatial approach that uncovers latent node representations by exploring high-order structural information within hypergraphs. Hyper-Atten (Bai, Zhang, and Torr 2019) introduced an attention mechanism to hypergraphs, enhancing their learning capabilities. UniGNN (Huang and Yang 2021) and HyperSAGE (Arya et al. 2020) take a direct message-passing approach on hypergraphs to learn representations. AllSet (Chien et al. 2021) has presented a powerful framework that unifies existing hypergraph learning methods.

In various fields like social networks (Sun et al. 2023), recommendation (Ding et al. 2023), and natural language processing (Xu et al. 2023a), hypergraphs have demonstrated their efficacy in tackling complex problems.

Datasets

We used four datasets, i.e. Christianity, Android, Douban and Memetracker, to conduct experiments.

Christianity (Sankar et al. 2020) consists of the user friendship network and cascading interactions related to Christian themes on Stack-Exchanges.

Android (Sankar et al. 2020) is collected from StackExchanges, which is a community Q&A website. It includes users' interactions across different channels, which form their friendship relations.

Douban (Zhong et al. 2012) is a Chinese social website where users can update their book reading statuses and follow the statuses of other users.

Memetracker (Leskovec, Backstrom, and Kleinberg 2009) collects a million news stories and blog posts from online websites, tracking the most frequent memes to analyze their migration among people. Each meme is considered an informational entity, while individual website URLs are treated as representations of users in the analysis.

Baselines

We compare thirteen representative baseline models with our models.

Macroscopic prediction models:

DeepCas (Li et al. 2016) transforms the cascade graph into node sequences through random walks and learns representations for each cascade using a deep learning framework.

DeepHawkes (Cao et al. 2017) integrates an end-to-end deep learning technique into the Hawkes process for cascade prediction.

CasCN (Chen et al. 2019b) applies GCN to capture the structures of information diffusion and uses LSTM to learn inherent dependencies between users' retweeting behaviors in sequential cascade information.

CasFlow (Xu et al. 2023b) leverages normalizing flows to learn node-level and cascade-level latent factors, enabling hierarchical pattern learning of information diffusion.

TCSE-net (Wu et al. 2022) preserves distinguishable structure patterns and eliminates potential noise by aligning and fusing temporal popularity and cascade information.

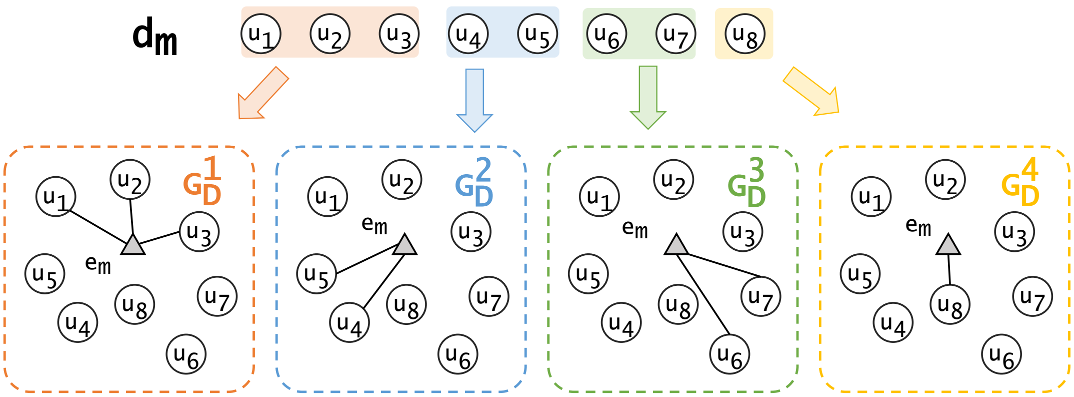


Figure 5: Hyperedge e_m connects different nodes in sequential hypergraphs.

Models	Android			Memetracker		
	Hits@100	**MAP@100**	**MSLE**	**Hits@100**	**MAP@100**	**MSLE**
w/o AdvDiff	0.2696	0.0711	0.467	0.5609	0.1605	0.895
w/o Diff	0.2758	0.0716	0.265	0.5747	0.1614	0.853
w/o Adv	0.2712	0.0718	0.369	0.5771	0.1634	0.844
w/o HGNN	0.5871	0.2013	1.074	0.3692	0.1178	0.581
w/o Macro	0.5580	0.1874	9.255	0.3665	0.1191	4.669
w/o Micro	0.5871	0.1937	0.865	0.3591	0.1174	0.711
MINDS	**0.2766**	**0.0727**	**0.151**	**0.5790**	**0.1638**	**0.506**

Table 6: Ablation study on Android and Memetracker datasets.

Microscopic prediction models:

TopoLSTM (Wang et al. 2017) extends the standard LSTM model to simulate the information diffusion process and combines it with the social network.

NDM (Yang et al. 2021) applies CNN to learn the diffusion representation of users and utilizes self-attention to make diffusion predictions.

SNIDSA (Wang, Chen, and Li 2018) explores diffusion paths and the social network to jointly learn heterogeneous information representations.

Inf-VAE (Sankar et al. 2020) embeds social homophily through GNNs and designs a co-attentive fusion network to integrate social and temporal variables.

DyHGCN (Yuan et al. 2020) jointly learns the structural characteristics of the social graph and dynamic diffusion graph, while encoding temporal information into a heterogeneous graph to capture users' dynamic preferences.

TAN-DRUD (Liu et al. 2022) models information cascades by capturing the dual role user dependencies of information senders and receivers.

Unified multi-scale prediction models:

FOREST (Yang et al. 2019) incorporates macroscopic information into an RNN-based microscopic diffusion model to simultaneously predict microscopic and macroscopic diffusion.

DMT-LIC (Chen et al. 2019a) designs a shared-representation layer to capture both the underlying structure of a cascade graph and the node sequence in the diffusion process.

Construction of Sequential Hypergraphs

In the case that the cascade d_m is divided into four parts based on the time periods, the nodes connected by the hyperedge e_m in the hypergraph corresponding to each time period are visually represented in Figure 5.

Supplementary Results to Ablation Study

We observe that the ablation study across two datasets in Table 5 may be insufficient. For example, *w/o AdvDiff* shows the best, worst, and average performance on three metrics compared to *w/o Adv* and *w/o Diff* respectively. To address this concern, we conducted ablation experiments on the other two datasets. The result is shown in Table 6. The suboptimal results on the Christianity and Douban datasets could be due to their unique characteristics, such as sparse network connections leading to minimal feature overlap.

Acknowledgements

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LDT23F01012F01 and Grant LDT23F01015F01, in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang Grant GK229909299001-008, in part by the National Natural Science Foundation of China under Grant 62372146, Grant 62071327 and Grant 61806061, and in part by the Zhejiang Laboratory Open Research Project under Grant K2022QA0AB01.

References

Arya, D.; Gupta, D. K.; Rudinac, S.; and Worring, M. 2020. HyperSAGE: Generalizing Inductive Representation Learning on Hypergraphs. arXiv:2010.04558.

Bai, S.; Zhang, F.; and Torr, P. H. S. 2019. Hypergraph Convolution and Hypergraph Attention. arXiv:1901.08150.

Cao, Q.; Shen, H.; Cen, K.; Ouyang, W. R.; and Cheng, X. 2017. DeepHawkes: Bridging the Gap between Prediction and Understanding of Information Cascades. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Cao, Q.; Shen, H.; Gao, J.; Wei, B.; and Cheng, X. 2019. Popularity Prediction on Social Platforms with Coupled Graph Neural Networks. *Proceedings of the 13th International Conference on Web Search and Data Mining*.

Chen, X.; Zhang, K.; Zhou, F.; Trajcevski, G.; Zhong, T.; and Zhang, F. 2019a. Information Cascades Modeling via Deep Multi-Task Learning. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Chen, X.; Zhou, F.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Zhang, F. 2019b. Information Diffusion Prediction via Recurrent Cascades Convolution. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 770–781.

Chien, E.; Pan, C.; Peng, J.; and Milenkovic, O. 2021. You are AllSet: A Multiset Function Framework for Hypergraph Neural Networks. arXiv:2106.13264.

Ding, C.; Zhao, Z.; Li, C.; Yu, Y.; and Zeng, Q. 2023. Session-based recommendation with hypergraph convolutional networks and sequential information embeddings. *Expert Systems with Applications*, 223: 119875.

Feng, S.; Cong, G.; Khan, A.; Li, X.; Liu, Y.; and Chee, Y. M. 2018a. Inf2vec: Latent Representation Model for Social Influence Embedding. *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 941–952.

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2018b. Hypergraph Neural Networks. In *AAAI Conference on Artificial Intelligence*.

Huang, J.; and Yang, J. 2021. UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks. arXiv:2105.00956.

Islam, M. R.; Muthiah, S.; Adhikari, B.; Prakash, B. A.; and Ramakrishnan, N. 2018. DeepDiffuse: Predicting the 'Who' and 'When' in Cascades. *2018 IEEE International Conference on Data Mining (ICDM)*, 1055–1060.

Jin, H.; Wu, Y.; Huang, H.; Song, Y.; Wei, H.; and Shi, X. 2022. Modeling Information Diffusion With Sequential Interactive Hypergraphs. *IEEE Transactions on Sustainable Computing*, 7: 644–655.

Ko, H.; Lee, S.; Park, Y.; and Choi, A. 2022. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, 11(1): 141.

Kong, S.; Mei, Q.; Feng, L.; Ye, F.; and Zhao, Z. 2014. Predicting bursts and popularity of hashtags in real-time. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*.

Leskovec, J.; Backstrom, L.; and Kleinberg, J. M. 2009. Meme-tracking and the dynamics of the news cycle. In *Knowledge Discovery and Data Mining*.

Li, C.; Ma, J.; Guo, X.; and Mei, Q. 2016. DeepCas: An End-to-end Predictor of Information Cascades. *Proceedings of the 26th International Conference on World Wide Web*.

Liu, B.; Yang, D.; Wang, Y.; and Shi, Y. 2022. Improving Information Cascade Modeling by Social Topology and Dual Role User Dependency. In *International Conference on Database Systems for Advanced Applications*.

Liu, P.; Qiu, X.; and Huang, X. 2017. Adversarial Multi-task Learning for Text Classification. In *Annual Meeting of the Association for Computational Linguistics*.

Miller, R.; and Lammas, N. 2010. Social media and its implications for viral marketing. *Asia Pacific Public Relations Journal*, 11(1): 1–9.

Sankar, A.; Zhang, X.; Krishnan, A.; and Han, J. 2020. Inf-VAE: A Variational Autoencoder Framework to Integrate Homophily and Influence in Diffusion Prediction. *Proceedings of the 13th International Conference on Web Search and Data Mining*.

Sun, L.; Rao, Y.; Zhang, X.; Lan, Y.; and Yu, S. 2022. MS-HGAT: Memory-Enhanced Sequential Hypergraph Attention Network for Information Diffusion Prediction. In *AAAI Conference on Artificial Intelligence*.

Sun, X.; Cheng, H.; Liu, B.; Li, J.; Chen, H.; Xu, G.; and Yin, H. 2023. Self-supervised hypergraph representation learning for sociological analysis. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, J.; Zheng, V. W.; Liu, Z.; and Chang, K. C.-C. 2017. Topological Recurrent Neural Network for Diffusion Prediction. *2017 IEEE International Conference on Data Mining (ICDM)*, 475–484.

Wang, S.; Hu, X.; Yu, P. S.; and Li, Z. 2014. MMRate: inferring multi-aspect diffusion networks with multi-pattern cascades. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Wang, Z.; Chen, C.; and Li, W. 2018. A Sequential Neural Information Diffusion Model with Structure Attention. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*.

Wang, Z.; and Li, W. 2019. Hierarchical Diffusion Attention Network. In *International Joint Conference on Artificial Intelligence*.

Wu, D.; Tan, Z.; Xia, Z.; and Ning, J. 2022. TCSE: Trend and cascade based spatiotemporal evolution network to predict online content popularity. *Multimedia Tools and Applications*, 82: 1459–1475.

Xu, H.; Zheng, C.; Zhao, Z.; and Sun, X. 2023a. Multi-Hypergraph Neural Networks for Emotion Recognition in Multi-Party Conversations. *Applied Sciences*, 13(3): 1660.

Xu, X.; Zhou, F.; Zhang, K.; Liu, S.; and Trajcevski, G. 2023b. CasFlow: Exploring Hierarchical Structures and Propagation Uncertainty for Cascade Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35: 3484–3499.

Yang, C.; Sun, M.; Liu, H.; Han, S.; Liu, Z.; and Luan, H. 2021. Neural Diffusion Model for Microscopic Cascade Study. *IEEE Transactions on Knowledge and Data Engineering*, 33: 1128–1139.

Yang, C.; Tang, J.; Sun, M.; Cui, G.; and Liu, Z. 2019. Full-Scale Information Diffusion Prediction With Reinforced Recurrent Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34: 2271–2283.

Yuan, C.; Li, J.; Zhou, W.; Lu, Y.; Zhang, X.; and Hu, S. 2020. DyHGCN: A Dynamic Heterogeneous Graph Convolutional Network to Learn Users' Dynamic Preferences for Information Diffusion Prediction. arXiv:2006.05169.

Zhang, Q.; Guo, Z.; Zhu, Y.; Vijayakumar, P.; Castiglione, A.; and Gupta, B. B. 2023. A deep learning-based fast fake news detection model for cyber-physical social services. *Pattern Recognition Letters*, 168: 31–38.

Zhao, Q.; Erdogdu, M. A.; He, H. Y.; Rajaraman, A.; and Leskovec, J. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Zhong, E.; Fan, W.; Wang, J.; Xiao, L.; and Li, Y. 2012. ComSoc: adaptive transfer of user behaviors over composite social network. In *Knowledge Discovery and Data Mining*.

Zhou, F.; Xu, X.; Zhang, K.; Trajcevski, G.; and Zhong, T. 2020. Variational Information Diffusion for Probabilistic Cascades Prediction. *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 1618–1627.