MS-HGAT: Memory-enhanced Sequential Hypergraph Attention Network for Information Diffusion Prediction

Ling Sun, Yuan Rao*, Xiangbo Zhang, Yuqian Lan, Shuanghe Yu

Xi'an Key Laboratory of Social Intelligence and Complexity Data Processing,
School of Software Engineering, Xi'an Jiaotong University, China
Shaanxi Joint Key Laboratory for Artifact Intelligence, China
{sunling, nick1001, Yuqian_Lan_xjtu, yushuanghe1997}@stu.xjtu.edu.cn, raoyuan@mail.xjtu.edu.cn

Abstract

Predicting the diffusion cascades is a critical task to understand information spread on social networks. Previous methods usually focus on the order or structure of the infected users in a single cascade, thus ignoring the global dependencies of users and cascades, limiting the performance of prediction. Current strategies to introduce social networks only learn the social homogeneity among users, which is not enough to describe their interaction preferences, let alone the dynamic changes. To address the above issues, we propose a novel information diffusion prediction model named Memory-enhanced Sequential Hypergraph Attention Networks (MS-HGAT). Specifically, to introduce the global dependencies of users, we not only take advantages of their friendships, but also consider their interactions at the cascade level. Furthermore, to dynamically capture users' preferences, we divide the diffusion hypergraph into several sub graphs based on timestamps, develop Hypergraph Attention Networks to learn the sequential hypergraphs, and connect them with gated fusion strategy. In addition, a memory-enhanced embedding lookup module is proposed to capture the learned user representations into the cascade-specific embedding space, thus highlighting the feature interaction within the cascade. The experimental results over four realistic datasets demonstrate that MS-HGAT significantly outperforms the state-of-the-art diffusion prediction models in both Hits@K and MAP@k metrics.

Introduction

New online social media allows people to access information in a cheap and handy way, thus facilitating rapid information sharing. Therefore, the information diffusion prediction technology, which aims to identify the potential users of information sharing, is urgently needed for addressing emerging scenarios in online social media, such as fake news controlling (Vosoughi, Roy, and Aral 2018; Wu et al. 2020b,a), hotspot detection (Yang et al. 2018) and online advertising (Liu et al. 2021).

Generally, information diffusion prediction methods can be divided into three categories: feature engineering-based methods, generative-based methods and representation learning-based methods. Among them, feature engineering-based methods usually extract representative features to predict the popularity of information propagation at the macro level (Cheng et al. 2014; Gao, Ma, and Chen 2014). However, they can hardly model the dependencies between users, nor can they capture the dynamic evolution of propagation structures. Probabilistic statistical generative models regarded information diffusion as event sequences in a time domain (Zhao et al. 2015; Bao 2016). However, this kind of approaches heavily depends on the predefined diffusion mechanism, while the propagation in real world may not strictly abide by the rule. In order to provide end-to-end solutions and improve the accuracy of information diffusion prediction, methods based on representation learning have been proposed. Prior works always focus on the structure or sequence of cascades, but ignore social structures that are not visible in cascades but have a significant impact on users' behaviour (Wang et al. 2017a,c; Li et al. 2017a), leaving them unable to predict the inactive users. Therefore, FOREST (Yang et al. 2019a) combines GRU and GCN to jointly learn the cascading contexts and social network. As the complete network structure is not always available, Inf-VAE (Sankar et al. 2020) embeds unobserved social connections by modeling homophily through variational autoencoder. Most recently, DyHGCN (Yuan et al. 2020) develops heterogeneous graphs to capture the users' interactions and social relationships jointly.

Although these improvements can partly describe the temporal influence and social homophily in cascades, they still suffer from some limitations. First, to introduce global dependencies of users, most of the existing methods take advantages of the friendship network, which is insufficient to describe users' interactive preferences and may even introduce noise. Second, user's preference changes dynamically, however, most of the existing methods ignore the dynamic connections between users and cascades, limiting their performance on prediction.

To address the above problems, we propose a **M**emory-enhanced **S**equential **H**ypergraph **A**ttention **N**etwork (MS-HGAT). Specifically, instead of learning the global dependencies of users from only the friendship network, we also construct hypergraphs to depict their interaction dependencies. Compared with the traditional GNNs, our proposed sequential HGATs can depict the dynamic interactions be-

*Corresponding author.

tween users and cascades through hyperedges and attention mechanism. Moreover, to capture the cascade level correlations, we design an extra aggregation operations in HGAT to preserve the features of cascades. In contrast to concat operation, the introduced gated fusion strategy controls the retaining rate of two vectors through correlation calculation, thus facilitating feature filtering. Besides, by designing memory-enhanced embedding look up and self-attention modules, MS-HGAT further emphasizes the interactions within the cascade to improve the prediction accuracy. The main contributions of this work are as follows:

- Enhance the global dependencies of users in information diffusion prediction: we take advantages of both the friendships and diffusion interactions of users to accurately predict the diffusion of information.
- Capture users' dynamic preferences for continuously prediction: we propose a sequential hypergraph attention network to learn the short-term interactions between users and cascades. The designed memory-enhanced embedding look up and self-attention module further emphasizes the feature evolution within the cascades.
- Experimental results demonstrate the effectiveness and robustness of MS-HGAT. Compared with the state-of-the-art diffusion prediction models, our model can achieve up to 6% improvement in Hits@100 score and 2% improvement in MAP@100 score.

Related Work

Information Diffusion Prediction

Information diffusion prediction amis to predict the future diffusion process based on the current cascades and relevant knowledge, such as social network structures (Li et al. 2017b). Cheng et al. (Cheng et al. 2014) analyzed diffusion processes from content, user, structural and temporal aspects, proved that all these features are helpful in prediction. Recently, since the feature engineering-based methods are not efficient in large-scale networks, and the generative methods mainly focus on process modeling but lack optimization in prediction, representation learning-based methods are proposed. Most previous studies learn user representation from the sequential or structured cascades with extended RNNs. For example, DeepDiffuse (Islam et al. 2018) utilized the time information through RNN and attention mechanism. TopoLSTM (Wang et al. 2017b) extended the standard LSTM model by allowing multiple inputs to model the cross dependency of cascades. SNIDSA (Wang, Chen, and Li 2018) introduced structure attention module and gating strategy to incorporate structural information into sequential information in RNN. However, none of them consider social relationships of users that do not manifest in cascades, which limit the accuracy of prediction. Therefore, FOREST (Yang et al. 2019a) and Inf-VAE (Sankar et al. 2020) embeded unobserved social connections to strengthen the prediction, while the learning based on static social networks is still too weak to capture users' dynamic interaction preferences. Inspired by Yuan et al.'s (Yuan et al. 2020) idea of utilizing global diffusion interactions, we innovatively introduce sequential hypergraphs to dynamically model users' preferences and integrate them with static social relations to optimize information diffusion prediction task.

Graph Neural Networks

The graph neural networks (GNNs) are widely used to learn non Euclidean graph structures, and have been proved to be effective in many tasks. The pioneer GNN (Scarselli et al. 2009) represents a node by exchanging its neighborhood information recurrently until convergence. Graph Convolutional Networks (GCNs) (Kipf and Welling 2017) generalize the convolution operation to graph data. Instead of training the embeddding vectors for each vertex, GraphSage (Hamilton, Ying, and Leskovec 2017) inductively trains a set of aggregators to accommodate the graph changes. Since the GNNs assume that all the neighbors of a node share the same weight during aggregation, it is not able to accurately model large noisy networks. Therefore, the Graph Attention Network (GAT) (Velickovic et al. 2018) specifies each neighbor a unique attention coefficient by multi-head attention mechanism. As a special graph, hypergraph contains hyperedges that join an arbitrary number of entities, which can naturally describe group relations in the real world. Recently, Feng et al. (Feng et al. 2019) proposed a hypergraph neural network using a Chebyshev expansion of the simple graph Laplacian. Ding et al. (Bai, Zhang, and Torr 2021) then introduced attention mechanism to hypergraph. Since group relationships such as co-authorship and co-participation are ubiquitous in the real world, hypergraphs have been used to solve problems in many fields, such as social networks (Yang et al. 2019b), recommendation (Wang et al. 2017b) and natural language processing (Ding et al. 2020). To the best of our knowledge, we are the first to apply hypergraph neural networks to information diffusion prediction task.

Problem Formulation

Since users' sharing behavior is always influenced by their personal interests and the external environment (Yang et al. 2019a), we firstly introduce the friendship graph and diffusion hypergraphs that are used for diffusion prediction in this paper. The friendship graph is represented as $G_F = (U, E)$, where U is the user set and E is the set of edges representing friendship. The historical diffusion cascades $C = \{c_1, c_2, ..., c_M\}, |C| = M$ are split into T subsets based on timestamps to construct diffusion hypergraphs $G_D = \{G_D^t | t = 1, 2, ..., T\}$, $G_D^t = (U^t, \mathcal{E}^t)$, where U^t is the user set and $\mathcal{E}^t$ represents hyperedges. Note that the node-hyperedge relationship of each hypergraph is distinct, that is, if u_i participates in c_m during the t-th time interval, the connection between u_i and e_m exists only in hypergraph $\mathcal{E}^t$, and is not visible in any other hypergraph.

Based on the above introductions, the information diffusion prediction task can be described as: given a user set $U = \{u_1, u_2, ..., u_n\}, |U| = N$, a friendship graph G_F, diffusion hypergraphs G_D and an observed diffusion sequence $c_m = \{(u_i^m, t_i^m) | u_i^m \in U\}$ (u_i^m referring the user u_i that activated by information m, t_i^m indicates the infection time), estimate the likelihood $\hat{y}_{u_j,m}$ that the user u_j will participate in c_m in the next step, and find out the next infected user by ranking the infection probabilities of all candidates.

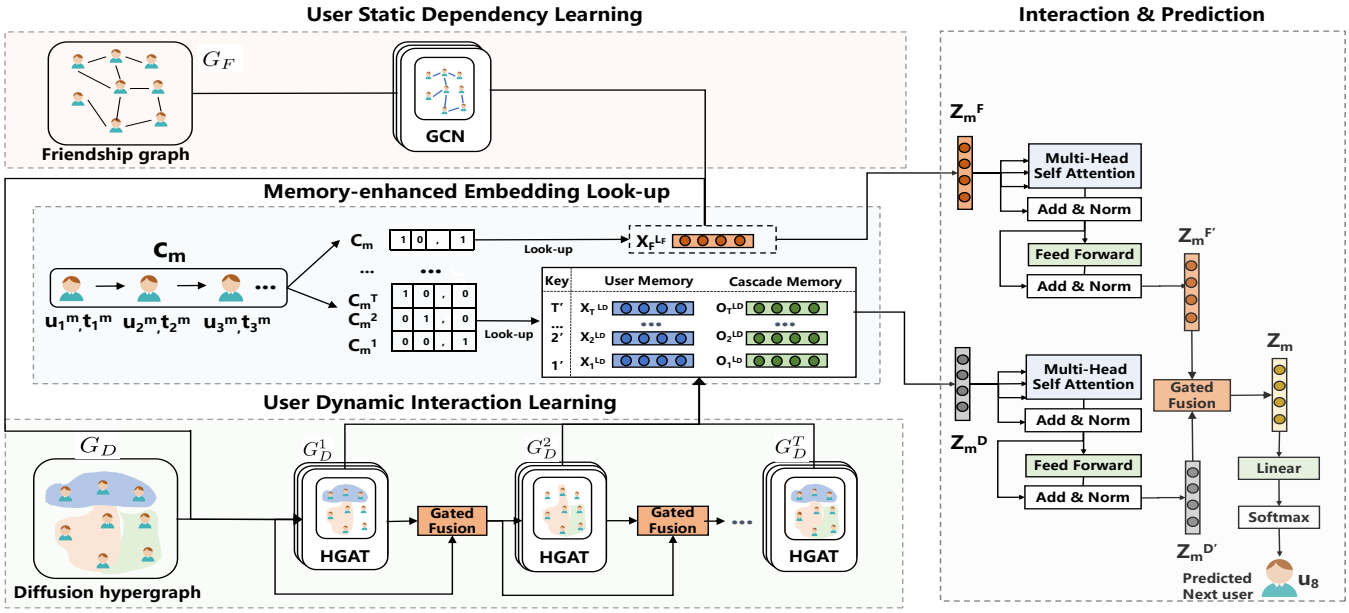


Figure 1: Four modules of MS-HGAT: 1) users' friendships are learned by GCN in user static dependency learning module; 2) user dynamic interaction learning module obtains interaction-based user and cascade embeddings through sequential HGATs; 3)memory-enhanced embedding look-up refers to finding the corresponding representation vectors in the static user representation and the dynamic memory block; 4)in interaction & prediction module, self-attention mechanisms are used to efficiently interact features in cascade, finally, probability of infection of candidates is calculated by Softmax function.

The Proposed Model

In this section, we introduce our **M**emory-enhanced **S**equential **H**ypergraph **A**ttention **N**etwork (MS-HGAT). The overall architecture of MS-HGAT is shown in Figure 1, which has four major components: 1) User static dependency learning module that learns users' friendships through GCN; 2) User dynamic interaction learning module, which obtains interaction-based user and cascade embeddings through sequential HGATs; 3) Memory-enhanced embedding look-up module, which captures the user representations into cascade-specific embedding space; 4) Interaction & prediction module, which learns interactive features within cascade and calculates infection probability of candidates. We introduce each component of our framework in detail in the following subsections.

User Static Dependency Learning

Users' static dependency can be expressed through friendship networks. The introduction of friendship network is conducive to user modeling, and can also alleviate the cold start problem in prediction, that is, even if a user has not participated in any cascade before, we can still learn its preference by exploring the characteristics of its neighbors. Considering that the structure of user friendship network is relatively stable, we assume it does not change in learning, and apply a multi-layer graph convolutional network (GCN) (Kipf and Welling 2017) to learn static user representation. Given friendship graph $G_F = (U, E)$, the layer-wise propagation rule of GCN can be defined as:

$$\mathbf{X}_F^{l+1} = \sigma\left(\tilde{\mathbf{D}_\mathbf{F}}^{-\frac{1}{2}}\tilde{\mathbf{A}_\mathbf{F}}\tilde{\mathbf{D}_\mathbf{F}}^{-\frac{1}{2}}\mathbf{X}_F^l\mathbf{W}_F\right) \tag{1}$$

where the initial user embeddings $\mathbf{X}_F^0 \in \mathbb{R}^{N\times d}$ is randomly initialized from normal distribution, d is the dimension of embedding, σ denotes the $\mathbf{ReLU}$ activation function, $\mathbf{W}_F$ is the trainable weight matrix, $\tilde{A_F} = A_F + I$ and $\tilde{D_F}$ are the adjacent and degree matrix of self-looped G_F, respectively. After being learned by a L_F-layer GCN, the static representation $\mathbf{X}_F^{L_F}$ of all users is finally obtained.

User Dynamic Interaction Learning

Since the user's friendships can not accurately reflect their interaction preferences, we construct sequential hypergraphs based on the cascades that have occurred before, and propose sequential hypergraph attention networks to dynamically learn user interactions at the cascade level, as well as the connections between cascades.

Hypergraph Attention Network (HGAT) At each time interval, we model the correlations among users through a hypergraph attention network (HGAT), the process of HGAT is shown in Figure 2.

Nodes-to-hyperedge aggregation. Given a hypergraph G_D^t, the first step of HGAT aims to learn the representation $o_{j,t}$ of hyperedge e_j^t by aggregating the initial user representaion $x_{i,t}$ of all its connected nodes u_i^t, formally:

$$\mathbf{o}_{j,t}^{l+1} = \sigma(\sum_{u_i^t \in e_j^t} \alpha_{ij}^t \mathbf{W}_1 \mathbf{x}_{i,t}^l) \tag{2}$$

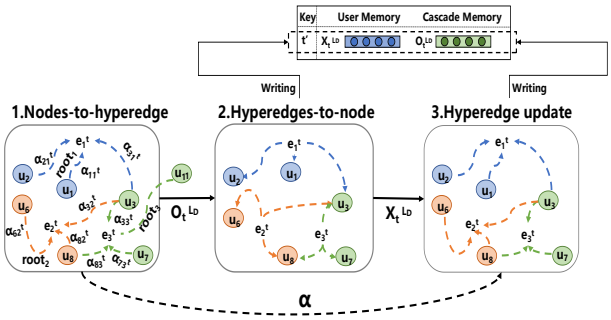


Figure 2: Stages of HGAT: (1) nodes-to-hyperedge aggregation, the root node of each cascade is determined in advance for attention calculation; (2) hyperedges-to-node aggregation, each hyperedge shares the same weight in this operation; (3) hyperedge update, the attention coefficient α is retained from first nodes-to-hyperedge aggregation. The learned user and the cascade representations are separately written to the memory module.

where σ is an activation function $\mathbf{ReLU}$, and $\mathbf{W}_1 \in \mathbb{R}^{d\times d}$ is the trainable weight matrix, d is the dimension of embedding. α_{ij}^t represents the attention coefficient of u_i^t in e_j^t.

Specifically, as we mentioned above, each hypergraph in our model only contains the user-cascade interaction information of the current time interval. This setting is designed to describe the short-term preference of users, but it will inevitably lead to information loss. Therefore, since the root node can partly reflect the content of the cascade, for each hypergraph, we preserve the information of roots of all hyperedges, and calculate the attention scores of other nodes by calculating the distance between them and the roots, that is:

$$\alpha_{ij}^t = \frac{\exp(-\operatorname{dis}(\mathbf{W}_1\mathbf{x}_{i,t}^l, \mathbf{W}_1\mathbf{r}_j^l))}{\sum_{u_p^t\in e_j^t}\exp(-\operatorname{dis}(\mathbf{W}_1\mathbf{x}_{p,t}^l, \mathbf{W}_1\mathbf{r}_j^l))} \tag{3}$$

where r_j^l is the representation of root user of e_j^t at layer l, and $\operatorname{dis}(\cdot)$ refers to the Euclidean distance.

Hyperedges-to-node aggregation. After obtaining representations of hyperedges, we train another aggregator to integrate all hyperedges $\mathcal{E}_i^t$ participated by u_i^t to learn the representation $x_{i,t}$of u_i during the t-th time interval, since all interactions between nodes and hyperedges occur in the same time interval, we give the same weight to each hyperedge when aggregating, that is:

$$\mathbf{x}_{i,t}^{l+1} = \sigma\Big(\sum_{e_j^t\in\mathcal{E}_i^t}\mathbf{W}_2\mathbf{o}_{j,t}^{l+1}\Big) \tag{4}$$

Update of hyperedges. The above two aggregations aim to learn users' interactions at the cascade level. In this process, hyperedges only play the role of feature transmission and are not well preserved. In order to further capture the connections between hyperedges, we design an extra step to update the hyperedge representation with all the embedding vectors of users, which can be denoted as:

$$\mathbf{o}_{j,t}^{l+1'} = \sigma\Big(\sum_{u_i^t\in e_j^t}\alpha_{ij}^t\mathbf{W}_3\mathbf{x}_{i,t}^{l+1}\Big) \tag{5}$$

Considering the time consumption of updating attention coefficients, we retained the coefficients α_{ij}^t used in the first Nodes-to-hyperedge aggregation.

Memory Module To leverage the features learned through HGATs at different time intervals, we construct a memory module M_D that stores not only the users, but also the cascade representation vectors in key-value form.

$$M_D = \left\{t' : (X_t^{L_D}, O_t^{L_D})\right\}, t = 1, 2, ..., T \tag{6}$$

where $X_t^{L_D} \in \mathbb{R}^{N\times d}$ and $O_t^{L_D} \in \mathbb{R}^{M\times d}$ are the user and cascade representations that learned from hypergraph G_D^t through a L_D-layer HGAT, t' corresponds to the maximum timestamp of interaction that occurs at the t-th time interval, T is the total number of intervals. The key-value pairs are stored in chronological order.

Sequential HGATs with Gated Fusion Strategy A single HGAT only learns the interactions in a short time interval, which can not accurately describe the dynamic changes of user preferences and cascade content with the evolution of diffusion. Therefore, we utilize multiple HGATs to learn the interactions in different time intervals, and strategically introduce a gated fusion strategy to connect them in chronological order, formally:

$$\begin{aligned}\mathbf{x}_{i,t+1}^0 &= g_{R_1}\mathbf{x}_{i,t}^{L_D} + (1-g_{R_1})\mathbf{x}_{i,t}^0\\ g_{R_1} &= \frac{\exp(\mathbf{W}_{Z_1}^T\sigma(\mathbf{W}_{R_1}\mathbf{x}_{i,t}^{L_D}))}{\exp(\mathbf{W}_{Z_1}^T\sigma(\mathbf{W}_{R_1}\mathbf{x}_{i,t}^{L_D}))+\exp(\mathbf{W}_{Z_1}^T\sigma(\mathbf{W}_{R_1}\mathbf{x}_{i,t}^0))}\end{aligned} \tag{7}$$

where $\mathbf{x}_{i,t}^0$ represents the initial embedding of user u_i^t and $\mathbf{x}_{i,t}^{L_D}$ is the output embedding for u_i^t, which is learned from G_D^t through a L_D-layer HGAT, $\mathbf{W}_{R_1}$ are the transformation matrix and $\mathbf{W}_{Z_1}^T$ is the vector for attention score calculation. $\sigma(\cdot)$ denotes the activation function $\mathbf{ReLU}$.

In order to improve the expression ability of HGAT and realize interactive learning, we use the user static representation learned from friendship network as the input of HGAT in the first time interval instead of initializing user representation from normal distribution, i.e., $\mathbf{X}_1^0 = \mathbf{X}_F^{L_F}$.

Memory-enhanced Embedding Look-up

In order to emphasize the representation interaction within the cascade, we further represent the target cascade by reading the embeddings of all its activated users from the static user representation and the dynamic memory module respectively in embedding look-up module, instead of directly using the last activated user.

Static representation look up. Given the target cascade $c_m = \{(u_i^m, t_i^m)|u_i^m \in U\}, i = 0, 1, ..., N-1$, we find the corresponding user embeddings x_i in static representation $X_F^{L_F}$ that is learned from friendship network, arrange them in the original order of the cascade, ignoring the specific timestamp, thus obtain $Z_m^F = [(x_i)] \in \mathbb{R}^{|c_m|\times d}, i = 0, 1, ..., N-1$.

Dynamic representation look up. Given the target cascade c_m, we query the corresponding users and cascades representations before the prediction timestamp from the memory module M_D respectively. In order to avoid information leakage, we read the user's representation at the nearest time interval before he or she participates in c_m, that is, supposing user u_i share the information m at time t_i^m, we first compare the value of t_i^m with the keys $[t']$ in memory module, if $t_i^m \geq t'$ and $t_i^m < (t+1)'$, u_i's embedding in $X_t^{L_D}$, i.e. $x_{i,t}$, will be read as its representation relative to cascade c_m. Hence, from the user perspective, c_m can be represented as $q_m^D = [(x_{i,t})] \in \mathbb{R}^{|c_m| \times d}, i = 0, 1, ..., N-1, t = 1, 2, ..., T$. Similarly, we use the same strategy to read the dynamic representation of the cascade in the memory module and obtain $p_m^D = [(o_{m,t})] \in \mathbb{R}^{|c_m| \times d}, t = 1, 2, ..., T$. Then the gated fusion mechanism is applied to integrate the cascade representation into users:

$$\begin{aligned} \mathbf{Z}_m^D &= g_{R_2}\mathbf{p}_m^D + (1 - g_{R_2})\mathbf{q}_m^D \\ g_{R_2} &= \frac{\exp(\mathbf{W}_{Z_2}^T \sigma(\mathbf{W}_{R_2}\mathbf{p}_m^D))}{\exp(\mathbf{W}_{Z_2}^T \sigma(\mathbf{W}_{R_2}\mathbf{p}_m^D)) + \exp(\mathbf{W}_{Z_2}^T \sigma(\mathbf{W}_{R_2}\mathbf{q}_m^D))} \end{aligned} \tag{8}$$

in which $\mathbf{W}_{R_2}$ and $\mathbf{W}_{Z_2}^T$ is the transformation matrix and vector for attention, $\sigma(\cdot)$ denotes activation function $\mathbf{tanh}$.

Feature Interaction & Prediction

Graph-based representation learning captures the co-occurrence relationship of users at the cascade level, but cannot further analyze the context interaction within the cascade. Therefore, based on the outstanding performance of self-attention layer in sequential tasks such as natural language processing, we apply two multi-head self-attention module to efficiently learn the static and dynamic feature interactions within cascades respectively, and obtain the final representation by post-fusion strategy for prediction.

Self-attention. Given the user static embeddings $Z_m^F = [(x_i)] \in \mathbb{R}^{|c_m| \times d}$, the sequential representation $\mathbf{h}_m^F$ is calculated as:

$$\begin{aligned} \text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d'}} + \mathbf{M}\right)\mathbf{V} \\ \mathbf{h}_{i,m}^F &= \text{Att}\left(\mathbf{Z}_m^F\mathbf{W}_i^Q, \mathbf{Z}_m^F\mathbf{W}_i^K, \mathbf{Z}_m^F\mathbf{W}_i^V\right) \\ \mathbf{h}_m^F &= \left[\mathbf{h}_{1,m}^F; \mathbf{h}_{2,m}^F; \ldots; \mathbf{h}_{H,m}^F\right]\mathbf{W}^O \end{aligned} \tag{9}$$

in which $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ and $\mathbf{W}^O$ are learnable transformation matrices, $d' = d/H$, d is the dimension of the embedding and H denotes the number of heads of attention. To avoid label leakage, we introduce a mask matrix $M \in \mathbb{R}^{|c_m| \times |c_m|}$ to block out future information, that is, $M_{i,j} = -\infty$ if $i > j$ else $M_{i,j} = 0$. Then, we obtain the attentive representaion $\mathbf{Z}_m^{F'}$ through a feed forward network (two layers fully-connected neural network):

$$\mathbf{Z}_m^{F'} = \mathbf{ReLU}\left(\mathbf{h}_m^F\mathbf{W}_{A_1} + \mathbf{b}_1\right)\mathbf{W}_{A_2} + \mathbf{b}_2 \tag{10}$$

where $\mathbf{W}_{A_1}$, $\mathbf{W}_{A_2}$ are learnable transformation matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias parameters. Similarly, we use another self-attention module to learn sequence $\mathbf{Z}_m^D$ and obtain $\mathbf{Z}_m^{D'}$, which represents the dynamic cascade context.

Fusion Layer. To incorporate both the static interaction $\mathbf{Z}_m^{F'}$ and dynamic interaction $\mathbf{Z}_m^{D'}$ of cascades for a more expressive representation $\mathbf{Z}_m$, we introduce a fusion layer:

$$\begin{aligned} \mathbf{Z}_m &= g_{R_3}\mathbf{Z}_m^{D'} + (1 - g_{R_3})\mathbf{Z}_m^{F'} \\ g_{R_3} &= \frac{\exp(\mathbf{W}_{Z_3}^T \sigma(\mathbf{W}_{R_3}\mathbf{Z}_m^{D'}))}{\exp(\mathbf{W}_{Z_3}^T \sigma(\mathbf{W}_{R_3}\mathbf{Z}_m^{D'})) + \exp(\mathbf{W}_{Z_3}^T \sigma(\mathbf{W}_{R_3}\mathbf{Z}_m^{F'}))} \end{aligned} \tag{11}$$

where $\mathbf{W}_{R_3}$ and $\mathbf{W}_{Z_3}^T$ are the transformation matrix and vector for attention. $\sigma(\cdot)$ denotes $\mathbf{tanh}$.

Diffusion Prediction Finally, we calculate the diffusion probabilities $\hat{y} \in \mathbb{R}^{|c_m| \times N}$ of users by:

$$\hat{y} = \text{softmax}(\mathbf{W}_p\mathbf{Z}_m + \mathbf{Mask}_m) \tag{12}$$

where $\mathbf{W}_p$ is a transformation matrix that maps the $\mathbf{Z}_m$ to user-specific space, $\mathbf{Mask}_m$ is used to mask users who have been activeted before prediction, that is, if the user u_i participates in c_m in step j, $(\mathbf{Mask}_m)_{1,i}^{j,i} = 0$, $(\mathbf{Mask}_m)_{j+1,i}^{|c_m|,i} = -\infty$. We adopt the cross entropy loss for training:

$$\mathcal{J}(\theta) = -\sum_{j=2}^{|c_m|}\sum_{i=1}^{|U|}\mathbf{y}_{ji}\log\left(\hat{\mathbf{y}}_{ji}\right) \tag{13}$$

in which θ represents all parameters that need to be learned in the model, if the user u_i participate in cascade c_m at the step j, $\mathbf{y}_{ji} = 1$, otherwise $\mathbf{y}_{ji} = 0$.

Experiments

To demonstrate the effectiveness of our proposed MS-HGAT model, we conduct extensive experiments on real datasets to answer the following research questions:

- **RQ1.** Can the proposed MS-HGAT outperform the state-of-the-art information diffusion prediction methods?
- **RQ2.** How do the quantity and quality of training sets affect the prediction performance of models?
- **RQ3.** How do the user relations and our learning strategies affect the prediction performance of MS-HGAT?

Experimental Setting

Datasets To explore the generalization of MS-HGAT, we sample data from both social platforms (Twitter and Douban) and Q&A websites (Android and Christianity). The details are listed in Table 1.

- **Twitter**(Hodas and Lerman 2013) contains tweets and its spreading paths among users during October 2010. We take the follow relation of users as friendship on Twitter.
- **Douban**(Zhong et al. 2012) is collected from a social website where users share their book or movie reading statuses and follow the statuses of others. The co-occurrence relation of users (e.g., read the same book) are taken as their friendship relation.
- **Android**.(Sankar et al. 2020) is collected from StackExchanges, a community Q&A websites. Users' interaction on various channels, such as questioning, answering constitutes their friendship relation.
- **Christianity**(Sankar et al. 2020) contains the user friendship network and cascading interactions related to Christian theme on the Stack-Exchanges.

Table 1: Statistics of datasets used in our experiments

Datasets	Twitter	Douban	Android	Christ.
# Users	12,627	12,232	9,958	2,897
		Frienship		
# Links	309,631	396,580	48,573	35,624
Density	24.52	30.21	4.87	12.30
		Interaction		
# Cascades	3,442	3,475	679	589
Avg. Length	32.60	21.76	33.3	22.9
Density	8.89	6.18	2.27	4.66

Evaluation Metrics According to the problem formulation, our prediction task can be regarded as a retrieval problem. Therefore, following the previous studies(Yang et al. 2019a; Sankar et al. 2020), we use two ranking metrics: Mean Average Precision on top k(MAP@k) and Hits score on top K (Hits@k) for model evaluation, $k = [10, 50, 100]$.

Baselines We compare MS-HGAT with the following information diffusion prediction methods:

- **DeepDiffuse**(Islam et al. 2018) uses RNN and attention mechanism to model the previously influenced users of cascades sequentially.
- **Topo-LSTM**(Wang et al. 2017b) extends the standard LSTM model to learn a dynamic directed acyclic graph, which contains structured diffusion information.
- **NDM**(Yang et al. 2021) models cascades through self-attention mechanism and CNNs, makes relaxed independence assumptions to alleviate long-term dependency.
- **SNIDSA**(Wang, Chen, and Li 2018) computes structural attention from diffusion path and learns sequential information of cascades through RNN.
- **FOREST**(Yang et al. 2019a) employs graph neural networks to learn users' social relationships and utilizes RNN to explore cascades context for prediction.
- **Inf-VAE**(Sankar et al. 2020) embeds social homophily through GNNs and designs a co-attentive fusion network to integrate the social and temporal variables.
- **DyHGCN**(Yuan et al. 2020) builds heterogeneous graphs which contain the social and diffusion relations of users, then learns user representation through GCN.

Parameter Settings Our experiments are conducted on a 12 GB GeForce GTX 2080Ti (GPU). For each dataset, we randomly choose 80% of the cascades for training, 10% are used for validation, and the remaining 10% are for test. The maximum cascade length is 200. For baselines, we preserve the settings as provided in original papers. For MS-HGAT, we implement it in PyTorch and adopt Adam as the optimizer, with a learning rate of 0.001. The dropout rate is 0.3, the batch size is 64, and the dimension of embedding is 64. We use a 2 layer GCN for friendship learning, and utilize a single layer HGAT for interaction learning since the designed three-step aggregation strategy is sufficient to capture higher-order interactions in a single hypergraph (comparison experiments are omitted due to space constraints). The number of subhypergraphs and self-attention heads is chosen from $[2 - 16]$, and set to be 8 and 14 after comparison.

Performance Comparison (RQ1)

We compare MS-HGAT with the baselines on four public datasets, the results are shown in Table 2 and 3. Specifically, we make the following observations:

First, MS-HGAT consistently achieves the best performances on all datasets. Compared to the second best model DyHGCN, MS-HGAT constructs a series of hypergraphs to dynamically describe the interactions between users and cascades, and uses memory and gating mechanism to store and fuse features effectively, thus reaching up to 6% improvement in Hits@100 score and 2% in MAP@100 score than DyHGCN. Second, the methods that exploit user social relations (SNIDSA, FOREST, Inf-VAE, DyHGCN and MS-HGAT) generally perform better than cascades-based approaches (DeepDiffuse, Topo-LSTM and NDM). Specifically, they achieve an average improvement of 13.98% and 5.43% on Hits and MAP indicators for all data sets, proving the validity of social relationship for prediction. Third, MS-HGAT and DyHGCN consider the global interaction relations of users and finally achieve the best results, which comfirms our assumption that users' interaction preferences can be learned from their historical behavior.

Impact of Training Set Proportion and Cascade Length (RQ2)

The performance of prediction may be affected by the quality of training sets. Therefore, we carry out comparative experiments on Twitter and Android dataset under different training proportion and cascade length to further prove the stability and validity of our model. Referring to Figure 3, we observe that our model can use only 60% of the data to achieve the performance of other models trained with 90%, which demonstrates the comprehensiveness of combining multiple relationships of users. Besides, Figure 4 shows that MS-HGAT achieves best performance under any length of cascade, proving the effectiveness of dynamically learning. The reason for the unstable performance on Android may be that both friendships and interactions of users in Android are relatively sparse (4.88 and 2.27, respectively), which is not enough to support the prediction of long cascades.

Ablation Study (RQ3)

We conduct ablation studies over the different parts of MS-HGAT on Twitter and Android datasets to investigate the contribution of submodules. The variants are designed as:

w/o FG removes social graph, i.e, $\mathbf{Z}_m = \mathbf{Z}_m^{D'}$ in Eq. 11.
w/o DH removes diffusion graphs, i.e, $\mathbf{Z}_m = \mathbf{Z}_m^{F'}$ in Eq. 11.
w/o UM ignores user memory, i.e, $\mathbf{Z}_m^D = \mathbf{p}_m^D$ in Eq. 8.
w/o CM ignores cascade memory, i.e, $\mathbf{Z}_m^D = \mathbf{q}_m^D$ in Eq. 8.
w/o ATTH ignores attention mechanism in HGATs, i.e, $\mathbf{o}_{j,t}^{l+1} = \sigma(\sum_{u_i^t \in e_j^t} \mathbf{W}_1 \mathbf{x}_{i,t}^l)$ in Eq. 2.
w/o GF replaces all gated fusions with concatenations.

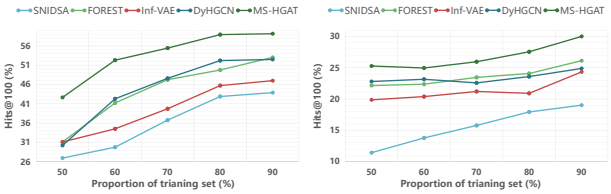
As shown in Table 4, MS-HGAT achieves the best performance compared to any of its variants, indicating the rationality of its design. Specifically, we observe that: First, model shows a significant decline after removing the social graph or diffusion hypergraphs, which proves the va-

Table 2: Experimental results on 4 dataset (%) (Hits@k scores for $K = 10, 50, 100$), scores are the higher the better.

Models	Twitter			Douban			Android			Christianity		
	@10	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
DeepDiffuse	5.79	10.80	18.39	9.02	14.93	19.13	4.13	10.58	17.21	10.27	21.83	30.74
Topo-LSTM	8.45	15.80	25.42	8.57	16.53	21.47	4.56	12.63	16.53	12.28	22.63	31.52
NDM	15.21	28.23	32.30	10.00	21.13	30.14	4.85	14.24	18.97	15.41	31.36	45.86
SNIDSA	25.37	36.64	42.89	16.23	27.24	35.59	5.63	15.22	20.93	17.74	34.58	48.76
FOREST	28.67	42.07	49.75	19.50	32.03	39.08	9.68	17.73	24.08	24.85	42.01	51.28
Inf-VAE	14.85	32.72	45.72	8.94	22.02	35.72	5.98	14.70	20.91	18.38	38.50	51.05
DyHGCN	31.88	45.05	52.19	18.71	32.33	39.71	9.10	16.38	23.09	26.62	42.80	52.47
MS-HGAT (ours)	**33.50**	**49.59**	**58.91**	**21.33**	**35.25**	**42.75**	**10.41**	**20.31**	**27.55**	**28.80**	**47.14**	**55.62**

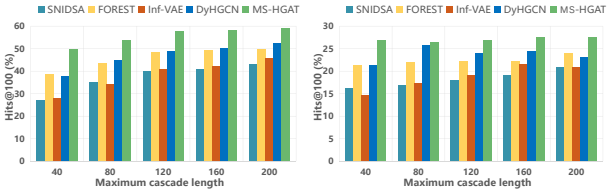
Table 3: Experimental results on 4 dataset (%) (MAP@k scores for $K = 10, 50, 100$), scores are the higher the better.

Models	Twitter			Douban			Android			Christianity		
	@10	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
DeepDiffuse	5.87	6.80	6.39	6.02	6.93	7.13	2.30	2.53	2.56	7.27	7.83	7.84
Topo-LSTM	8.51	12.68	13.68	6.57	7.53	7.78	3.60	4.05	4.06	7.93	8.67	9.86
NDM	12.41	13.23	14.30	8.24	8.73	9.14	2.01	2.22	2.93	7.41	7.68	7.86
SNIDSA	15.34	16.64	16.89	10.02	11.24	11.59	2.98	3.24	3.97	8.69	8.94	9.72
FOREST	19.60	20.21	21.75	11.26	11.84	11.94	5.83	6.17	6.26	14.64	15.45	15.58
Inf-VAE	19.80	20.66	21.32	11.02	11.28	12.28	4.82	4.86	5.27	9.25	11.96	12.45
DyHGCN	20.87	21.48	21.58	10.61	11.26	11.36	6.09	6.40	6.50	15.64	16.30	16.44
MS-HGAT (ours)	**22.49**	**23.17**	**23.30**	**11.72**	**12.52**	**12.60**	**6.39**	**6.87**	**6.96**	**17.44**	**18.27**	**18.40**



(a) Twitter (b) Android

Figure 3: Impact of training proportion.



(a) Twitter (b) Android

Figure 4: Impact of maximum cascade lengths.

Table 4: Ablation study of MS-HGAT.

Models	Twitter		Android	
	Hits@100	**MAP@100**	**Hits@100**	**MAP@100**
MS-HGAT	58.91	**23.30**	**27.55**	**6.96**
w/o FG	57.20	21.38	26.32	6.86
w/o DH	57.41	22.24	26.74	6.78
w/o UM	58.63	22.74	26.40	6.83
w/o CM	58.32	21.96	27.09	6.77
w/o ATTH	**58.95**	22.76	27.03	6.75
w/o GF	57.93	22.19	27.26	6.89

lidity of introducing these two types of global dependencies. Second, the memory module retains the dynamic interactions of user-user and cascade-cascade, respectively. Together with the lookup operation, it helps to improve the performance of prediction by capturing the learned interactions into cascade-specific space. Third, the gated fusion and attention strategy determines the importance of vectors through correlation calculation, thus facilitating feature filtering for prediction. It is noted that even though attention mechanism improves the model's performance in most cases, MS-HGAT without attention has a slightly higher Hits@100 value on Twitter. The reason may be that the learning of users' intimate relationships in Twitter can effectively describe their behavior (each user has an average of 24.52 friends and 8.89 participation cascades, compared with 4.88 and 2.27 for Android), in this case, the introduction of distant root dependencies may not help much. According to the improvements shown on Android, we conclude that attention mechanism is more conducive to cold start scenario.

Conclusion

In this work, we propose a novel memory-enhanced sequential hypergraph attention network (MS-HGAT) for information diffusion prediction, which jointly learns users' social and diffusion relationships. Through the learning of GCN, sequential HGATs and self-attention module, our model fully and dynamically captures the interactions from the aspects of user-user, cascade-cascade and user-cascade. The experimental results demonstrate the effectiveness of MS-HGAT. In the future, we consider using hypergraph to describe the tree-shaped cascades, not just the sequences, and combine the information content to improve the prediction.

Acknowledgments

The research work is supported by National Key Research and Development Program in China (2019YFB2102300); The World-Class Universities (Disciplines) and the Characteristic Development Guidance Funds for the Central Universities (PY3A022); Ministry of Education Fund Projects (No. 18JZD022 and 2017B00030); Shenzhen Science and Technology Project (JCYJ20180306170836595); Basic Scientific Research Operating Expenses of Central Universities (No.ZDYF2017006); Xi'an Navinfo Corp.& Engineering Center of Xi'an Intelligence Spatial-temporal Data Analysis Project (C2020103); Beilin District of Xi'an Science & Technology Project (GX1803). We would like to thank the reviewers for their time and constructive comments.

References

Bai, S.; Zhang, F.; and Torr, P. H. S. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognit.*, 110: 107637.

Bao, P. 2016. Modeling and Predicting Popularity Dynamics via an Influence-based Self-Excited Hawkes Process. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*, 1897–1900.

Cheng, J.; Adamic, L.; Dow, A.; Kleinberg, J.; and Leskovec, J. 2014. Can Cascades be Predicted? *Proceedings of the 23rd International Conference on World Wide Web, WWW 2014*.

Ding, K.; Wang, J.; Li, J.; Li, D.; and Liu, H. 2020. Be More with Less: Hypergraph Attention Networks for Inductive Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 4927–4936. Association for Computational Linguistics.

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 3558–3565. AAAI Press.

Gao, S.; Ma, J.; and Chen, Z. 2014. Effective and effortless features for popularity prediction in microblogging network. 269–270.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 1024–1034.

Hodas, N. O.; and Lerman, K. 2013. The Simple Rules of Social Contagion. *CoRR*, abs/1308.5015.

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

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Li, C.; Ma, J.; Guo, X.; and Mei, Q. 2017a. DeepCas: An End-to-end Predictor of Information Cascades. In *Proceedings of the 26th International Conference on World Wide Web,WWW 2017*, 577–586.

Li, D.; Zhang, S.; Sun, X.; Zhou, H.; Li, S.; and Li, X. 2017b. Modeling Information Diffusion over Social Networks for Temporal Dynamic Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 29(9): 1985–1997.

Liu, Y.; Yang, S.; Zhang, Y.; Miao, C.; Nie, Z.; and Zhang, J. 2021. Learning Hierarchical Review Graph Representations for Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 1–1.

Sankar, A.; Zhang, X.; Krishnan, A.; and Han, J. 2020. Inf-VAE: A Variational Autoencoder Framework to Integrate Homophily and Influence in Diffusion Prediction. In *Proceedings of The 12th ACM International Conference on Web Search and Data Mining, WSDM2020*, 510–518.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Vosoughi, S.; Roy, D.; and Aral, S. 2018. The spread of true and false news online. *Science*, 359(6380): 1146–1151.

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

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

Wang, Y.; Shen, H.; Liu, S.; Gao, J.; and Cheng, X. 2017c. Cascade Dynamics Modeling with Attention-based Recurrent Neural Network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, 2985–2991.

Wang, Z.; Chen, C.; and Li, W. 2018. A Sequential Neural Information Diffusion Model with Structure Attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 1795–1798. ACM.

Wu, L.; Rao, Y.; Yang, X.; Wang, W.; and Nazir, A. 2020a. Evidence-Aware Hierarchical Interactive Attention Networks for Explainable Claim Verification. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 1388–1394.

Wu, L.; Rao, Y.; Zhao, Y.; Liang, H.; and Nazir, A. 2020b. DTCA: Decision Tree-based Co-Attention Networks for Explainable Claim Verification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 1024–1035.

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(3): 1128–1139.

Yang, C.; Tang, J.; Sun, M.; Cui, G.; and Liu, Z. 2019a. Multi-scale Information Diffusion Prediction with Reinforced Recurrent Networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI2019*, 4033–4039.

Yang, C.; Wu, Q.; Gao, X.; and Chen, G. 2018. EPOC: A Survival Perspective Early Pattern Detection Model for Outbreak Cascades. In Hartmann, S.; Ma, H.; Hameurlain, A.; Pernul, G.; and Wagner, R. R., eds., *Proceedings of Database and Expert Systems Applications - 29th International Conference, DEXA 2018*, volume 11029 of *Lecture Notes in Computer Science*, 336–351.

Yang, D.; Qu, B.; Yang, J.; and Cudré-Mauroux, P. 2019b. Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2147–2157. ACM.

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 preprint arXiv:2006.05169*.

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. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,KDD 2015*, 1513–1522.

Zhong, E.; Fan, W.; Wang, J.; Xiao, L.; and Li, Y. 2012. ComSoc: adaptive transfer of user behaviors over composite social network. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, 696–704. ACM.