

## Research Article

# A Hierarchical Attention Recommender System Based on Cross-Domain Social Networks

Rongmei Zhao ,<sup>1</sup> Xi Xiong ,<sup>1,2</sup> Xia Zu ,<sup>3</sup> Shenggen Ju,<sup>4</sup> Zhongzhi Li,<sup>1</sup> and Binyong Li<sup>1</sup>

<sup>1</sup>School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China

<sup>2</sup>School of Aeronautics and Astronautics, Sichuan University, Chengdu 610065, China

<sup>3</sup>College of Computer Science, Sichuan University, Chengdu 610065, China

<sup>4</sup>School of Management, Chengdu University of Information Technology, Chengdu 610103, China

Correspondence should be addressed to Xia Zu; [sunnyzu@163.com](mailto:sunnyzu@163.com)

Received 14 June 2020; Revised 6 July 2020; Accepted 23 July 2020; Published 12 August 2020

Guest Editor: Hongshu Chen

Copyright © 2020 Rongmei Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Search engines and recommendation systems are an essential means of solving information overload, and recommendation algorithms are the core of recommendation systems. Recently, the recommendation algorithm of graph neural network based on social network has greatly improved the quality of the recommendation system. However, these methods paid far too little attention to the heterogeneity of social networks. Indeed, ignoring the heterogeneity of connections between users and interactions between users and items may seriously affect user representation. In this paper, we propose a hierarchical attention recommendation system (HA-RS) based on mask social network, combining social network information and user behavior information, which improves not only the accuracy of recommendation but also the flexibility of the network. First, learning the node representation in the item domain through the proposed Context-NE model and then the feature information of neighbor nodes in social domain is aggregated through the hierarchical attention network. It can fuse the information in the heterogeneous network (social domain and item domain) through the above two steps. We propose the mask mechanism to solve the cold-start issues for users and items by randomly masking some nodes in the item domain and in the social domain during the training process. Comprehensive experiments on four real-world datasets show the effectiveness of the proposed method.

## 1. Introduction

With the acceleration of people's daily life, much time can be saved to get useful information quickly in a practical way, and recommender systems play a critical role in filtering information. Collaborative filtering recommendation [1] is the mainstream traditional recommendation algorithm. However, due to the data sparsity, it seriously affects the recommendation quality. The collaborative filtering recommendation combined with the neural network [2] (e.g., CNN, RNN, and CDAE) alleviates this problem. Besides, taking advantage of the productive relationship in the social networks [3–5] can effectively solve the cold-start problem [6, 7], but there is a malicious fraud problem by distrusting users.

On one hand, in social networks, there are explicit or implicit relationships between most users, which influence

each other's behaviors. However, not every user and connection in a social network is trustworthy. To reduce the impact of untrusted users, social networks can be divided into many small ego-networks (representing all the users both directly and indirectly trusted by  $u$ ) according to users' trust. Adding a trusted network into the recommender system can effectively solve the problem of a fraud attack. The traditional attribute graph-based semisupervised classification methods, such as label propagation (LP) [8] and label spreading (LS) [9], can effectively classify the unstructured data and have been used in community (consisting of similarity nodes) detection. In [10], they propose a new method to extend the global reputation model with local reputation, through an unsupervised process calculated in the user's ego-network. On the other hand, the widespread existence of social media has dramatically enriched the social

activities of network users and produced productive social relationships; the integration of these social relationships has improved the quality of the recommender system and solved the cold-start problem. In recent years, researchers have proposed a large number of recommender systems [11, 12] based on social networks that homogenize social relationships. However, it may restrain user representation learning in each respective domain, since users behave and interact differently in two domains, which makes their representations heterogeneous [13]. Some research [14, 15] considers the heterogeneity, and learning separated latent representations and then transferred them from a source domain to a target domain for a recommendation. However, learning the representations is challenging due to the inherent data sparsity problem in both domains.

Based on the latest advances in deep learning, especially the development of graph convolutional networks (GCN) [16], it is more accessible to aggregate features in social networks. The recommender system proposed by researchers based on graph convolutional neural network [17] is superior to the previous recommendation algorithms in recommendation quality; Chebyshev [18] is a classical graph convolution based on the spectral method. However, graph convolution assigns the same weight to neighbor nodes, and the feature aggregation depends on the entire graph, which limits the flexibility and generalization of the graph. Therefore, a graph attention network [19] is proposed, which uses the attention mechanism to weigh the sum of the features of neighboring nodes, and the feature weights of neighboring nodes are related to the nodes themselves, whether it is a graph convolutional neural network or a graph attention network, processing the user nodes and item nodes in the same way, without considering the heterogeneity of the two types of nodes. Besides, the training time is longer when the network is more massive.

In this paper, we consider the heterogeneity of information and relationships in the social and item domains, propose the Context-NE model to learn the representation in the item domain. We can learn representations of users interacting with multiple items through the model. The context contains the basic properties of the node and the user's comments on the item so that it can fuse the structure information and text information in the item domain. Besides, based on the graph convolutional networks, we propose a hierarchical attention mechanism and mask mechanism, which can improve the quality of recommendations and the generalization ability and flexibility of the network. Our significant contributions can be summarized as follows:

- (i) We propose a network embedding method Context-NE, which learns the node embedding in the item domain. It mainly represents the information of multiple items visited by the user and obtains the user's node representation in this domain.
- (ii) We use the hierarchical attention mechanism to aggregate the node information in the social domain. At first, we use the K-head self-attention mechanism the same as GAT to aggregate the

information of neighbor nodes; then, we use the attention mechanism to aggregate the output information of the K-head self-attention mechanism to obtain the final representation of the node.

- (iii) We apply the mask mechanism during training and processing the features of a mask node according to the mask mechanism. Newly joined nodes (e.g., new users and new items) can be regarded as a masked node, increasing the flexibility and generalization of the network.
- (iv) We conducted extensive experiments to evaluate the performance of the proposed HA-RS model on citation networks and Yelp datasets, and the results showed that our approach performed better than the most advanced baseline approaches.

The remainder of this paper is structured as follows: Section 2 discusses the related works. Section 3 gives preliminary definitions in the paper and gives a description of the problem. Section 4 introduces the method of hierarchical attention recommender system based on cross-domain social network, which is divided into four parts. The experimental results are demonstrated in Section 5. Lastly, the full study is concluded in Section 6.

## 2. Related Works

In this section, we will introduce the study of the traditional recommender system, trust recommender system, and graph convolution recommender system, respectively. This paper builds on the latest advanced methods based on social networks.

**2.1. Traditional Recommender Systems.** It is mainly divided into two categories. One is content-based recommender systems [20] which use content information of users and items, such as their respective occupation and genre, to predict the next purchase of a user or rating of an item. The other is collaborative filtering models [21] which find the choices of users by mining the historical behavior data of users, divide the groups of users based on different preferences, and recommend commodities with similar interests. The traditional the recommendation algorithm has serious cold-start problems.

**2.2. Trust Recommender System.** Social network-based recommender systems [22] can mitigate the cold-start problem, and most of the information on social networks is unreliable due to different backgrounds or preferences of users. The trust network is introduced into the recommender system to solve the problem of a fraud attack. Yang et al. [23] propose a hybrid method (TrustMF) that combines both a trustor model and a trustee model from the perspectives of trustors and trustees; that is, both the users who trust the active user and those who are trusted by the user will influence the user's ratings on unknown items. Guo et al. [24] propose TrustSVD, a trust-based matrix factorization technique, which takes into account both the explicit and implicit

influence of ratings and trust information when predicting ratings of unknown items. Pasquale et al. [10] propose a model capable of integrating local and global reputation in online social networks and calculating the user's ego-network through an unsupervised method. In this paper, it claims that the local reputations generally perform better than global ones, although they could not always be sufficient for calculating reliable values.

**2.3. Graph Convolution Recommender System.** Topological relations [25] and information transmission [26] between nodes in social networks are very important for social recommendation. Therefore, fully extracting the information of network structure is one of the key parts of recommendation algorithm. Based on the graph embedding method and drawing on the ideas of CNN, a graph neural network (GNN) [27] is proposed to collect information from the graph structure. GNN is a compelling, structured data modeling architecture; however, GNN uses the same parameters in the iteration and cannot aggregate edge information. To solve some of the limitations of graph neural networks, some variants of graph neural networks (e.g., GCN, DCNN, GAT, and GGNN) have gradually appeared. Instead of directly propagating all the attributes of each node in GCN and GAT, Masked GCN [28] only propagates a portion of its attributes to the neighbors. PinSage [29] combines efficient random walks and graph convolutions to generate embeddings of nodes (i.e., items) that incorporate both graph structure and node feature information. STAR-GCN [30] can produce node embeddings for new nodes by reconstructing masked input node embeddings, which primarily tackles the cold-start problem.

Despite the compelling success achieved by these methods in general cases, they ignore the heterogeneity of social networks. It may restrain user representation learning in each respective domain since users behave and interact differently in social networks. GraphRec [31] and DiffNet [32] consider the heterogeneity of social network, but they separately learn a latent representation of users and items. Poor network flexibility and scalability is a common problem in graph-based recommendation algorithms. The main concern of this paper is to consider the heterogeneity of social networks and improve the flexibility of networks.

### 3. Definitions and Problem Statement

The social network used in this paper is a heterogeneous graph, which contains multiple types of nodes (users and item nodes), as well as various types of edges (e.g., the relationship between users and the interaction between users and items).

**Definition 1.** Social domain and item domain: we divide the social network into the social domain and the item domain, which represent the user-user connections and user-item interactions.  $I = \{I_1, I_2, \dots, I_m\}$  represents the item sets and  $m$  indicates that there are  $m$  items in the item domain. The user sets are represented by  $U = \{U_1, U_2, \dots, U_n\}$  and  $n$

represents the total of users in the social domain. The edges between nodes represent friend relationships are represented by  $F$ ; the edges between the user and the item represent their interaction and are represented by  $T$ . The entire network can be represented through a quaternary  $G = \{U, B, F, T\}$  and the overall network is shown in Figure 1.

**Definition 2.** User embedding and item embedding: item embedding contains information such as the user's reviews of the item, the score, and the attributes of the item (e.g., item name and type). The initial user representation includes the primary attributes of the user (e.g., name and age).

**Definition 3.** Social homogeneity and item homogeneity: take the user  $U_1$  as an example, the user directly connected to the user is a first-order neighbor (e.g.,  $U_2$  and  $U_3$ ), the user directly connected to the first-order neighbor is a second-order neighbor, and so on. User  $U_1$  tends to choose items  $I_1$  and  $I_2$  that first-order neighbors and second-order neighbors prefer (i.e., users and friends have similar preferences), which is called social homogeneity. User  $U_1$  also tends to choose item  $I_6$  which is similar to item  $I_5$  selected by the user's historical behavior, which is called item homogeneity. We will further introduce item homogeneity and social homogeneity in Sections 4.1 and 4.2, respectively.

**Definition 4.** Network mask: we randomly selected some nodes in network  $G$  and masked some attributes of these nodes. We describe the mask mechanism in detail in Section 4.3; the users ( $U_3$  and  $U_4$ ) and the items ( $I_1$  and  $I_6$ ) are the masked nodes in the network  $G$ .

**3.1. Task Description.** We divided the heterogeneous network into the social domain and item domain. Firstly, we learn the user representation with user preference in the item domain and then aggregate the social influence in the user domain. Finally, we use the user representation to recommend items.

### 4. Model

The HA-RS model framework is shown in Figure 2 and mainly includes two parts, and one is the user's node representation model Context-NE in the item domain; it can be seen on the left side of Figure 2. The other is the hierarchical attention embedding model, which is used to learn the feature of user friends; it can be seen on the right side of Figure 2. The following content will describe the model in detail.

**4.1. Context-NE.** Considering the item homogeneity, users tend to choose the item similar to the historical item; that is, the user's past behavior will affect the user's next behavior. We obtain the user representation in item domain through the Context-NE model, which contains the historical preference of the user. The Context-NE

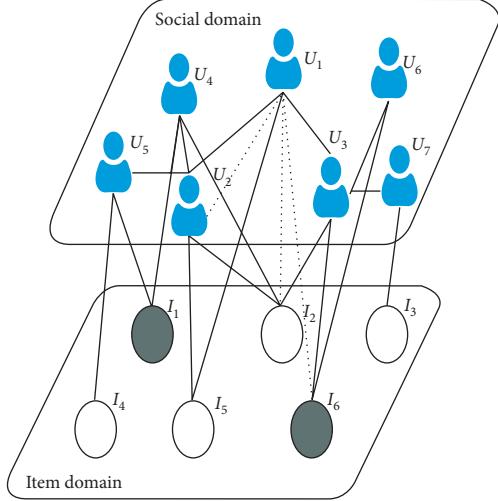


FIGURE 1: Hierarchical representation of heterogeneous social networks. The users ( $U_3$  and  $U_4$ ) and the items ( $I_1$  and  $I_6$ ) are the masked nodes in the network  $G$ .

model learns the user embedding in the item domain, and we call the information of users/items (refers to the point of interest) as a context in this paper. The user's context includes the primary attributes of the user,  $\text{context} = \{\text{user'id, user'name, friend'name, fans'number}\}$ . The context of the item includes the user's review of the item and the primary attributes of the item,  $\text{context} = \{\text{review, bussiness'id, bussiness'name, city, latitude, longitude, categories}\}$ . Considering the interactive sparse of the user in the item domain, we use LSTM to learn context sequence vector other than the score matrix. The Context Description transforms each word in the sequence into the corresponding word vector  $w$  through the LSTM model. It then merges context information (in particular, the context-dependence of the review) into the sequence embedding  $S$ ,  $S = \{w_1, w_2, w_d\}, S \in R^d$ . The Convolution Layer extracts local features from the sequence embedding  $S$ , the number of convolution kernels is  $M$ , the size is  $C$ , and the size of the sliding window is  $L_w$ . The user context vector and item context vector after convolution are expressed as  $\vec{U}_c$  and  $\vec{I}_c$ , respectively. Calculation formula is as shown in the following equations:

$$\vec{U}_c = \text{Conv1}(K_m, S^u) + b_m, \quad (1)$$

$$\vec{I}_c = \text{Conv1}(K_m, S^i) + b_m, \quad (2)$$

where  $K_m$  is the  $m$ -th convolution kernel,  $b_m$  is bias,  $S^u$  represents the sequence embedding of the user, and  $S^i$  represents the sequence embedding of the item.

The pooling layer uses mean pooling to reduce the output dimension and prevent overfitting. The sliding window size is also  $L_w$ . The user context vector and item context vector after pooling are expressed as  $\vec{U}_i$  and  $\vec{I}_{ij}$ , respectively, and the calculation formula is as shown in the following equations:

$$\vec{U}_i = \text{meanpool}(\vec{U}_c), \quad (3)$$

$$\vec{I}_{ij} = \text{meanpool}(\vec{I}_c). \quad (4)$$

We get the user representation of the  $i$ -th user in the item domain after the user embedding and the item embedding are nonlinearized by the Tanh function, which is represented by  $\vec{Z}_i$ ; the calculation formula is as shown in the following equation:

$$\vec{Z}_i = r_j \text{Tanh}(\vec{U}_i^T W_{ij} \vec{I}_{ij} + b_{ij}), \quad (5)$$

where  $\vec{I}_{ij}$  represents the embedding of item  $j$  that user  $i$  has visited,  $\vec{U}_i$  represents the representation of user  $i$ , and  $W_{ij}$  and  $b_{ij}$  represent weight parameters and bias parameters, respectively.  $r_j$  represents the user's preference for each item,  $\text{check}_j$  represents the number of interactions between user  $i$  and item  $j$ , and  $\text{check}_i$  represents the total number of interactions between user  $i$  and items. The more times user  $i$  interacts with the same item, the more the user likes the item; that is, the larger the  $r_j$  value, the greater the proportion of the item embedded in the user representation.  $r_j$  can be calculated by the following equation:

$$r_j = \frac{\text{check}_j}{\text{check}_i}. \quad (6)$$

**4.2. Hierarchical Attention.** In the social domain, hierarchical attention mechanism and graph convolution are mainly used to fuse the feature information of user neighbors into user embedding. The user representation finally obtained contain the user's preference information and the social impact in social networks, and utilizing it to recommend will be closer to the user's preferences.

Learning the user representation in the social domain should consider the friend relationships, that is, neighbor nodes in the network graph. However, it is not worth calculating all neighbor nodes in large social networks. Every user has too many neighbors, but not every neighbor has a significant impact on the user's preferences. For example, user  $A$  has a neighbor  $C$  that infrequently interacts; the impact of  $C$  on user  $A$  will be minimal. Besides, we will consider  $L$ -layer neighbor in the aggregating process, if we aggregate the features of all the neighbor nodes that will increase the noise in the user's representation. Meanwhile, the time cost of calculating all  $L$ -layer neighbor nodes in a large-scale network is very high.

We prune the neighbor node in this paper. First, we calculate the node similarity between the user and the neighbor and, then, sort and select the node of the top  $T$  as the user's neighbor when the number of neighbors of the user is higher than  $T$ ; otherwise, there is no need to delete the neighbor, and the pruning strategy can be expressed as the following equation:

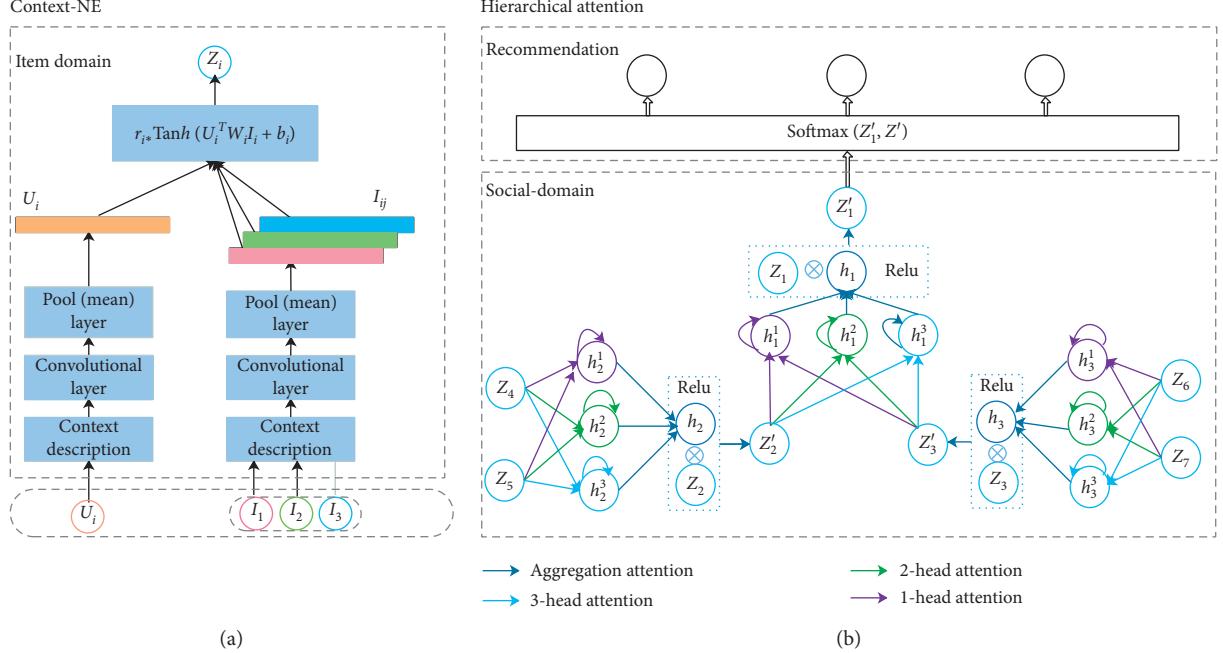


FIGURE 2: The framework of HA-RS model in this paper. (a) The user's node representation model Context-NE in the item domain. (b) The hierarchical attention embedding model, which is used to learn the feature of user friends.

$$N(u) = \begin{cases} N(u), & \text{if } 0 < |N(u)| < T, \\ \text{top}_T[f(u, v)], & \text{else,} \end{cases} \quad (7)$$

where  $\text{top}[\cdot]$  means take  $T$  users ( $v$ ) with the highest similarity to user  $u$ ,  $f(u, v) = u^T v$  is the similarity calculation function between two nodes, and  $N(u)$  represents the neighbors of node  $u$ . Only considering the more critical neighbor nodes can reduce the computation and improve the quality of the user representation, which can be applied to large-scale networks.

In this paper, we assume that users' preferences are related to their  $L$ -order neighbors and use the graph convolution network to aggregate the features of the user's neighbor. Specifically, we assign different weights to the user's neighbors through the hierarchical attention mechanism to indicate the influence degree of different neighbors on users. This process can be likened to the calculation of local reputation in ego-network, where users'  $L$ -order neighbors are user friends in ego-network and the function of attention coefficient is similar to the local reputation. The input of graph convolution network is the embedded set  $Z$  learned in the item domain,  $Z = \{Z_1, Z_2, \dots, Z_N\}$ ,  $Z_i \in R^F$ ,  $N$  represents the number of user nodes, and  $F$  represents the number of features in each node. The output through the layer attention mechanism is represented as  $Z'$ ,  $Z' = \{Z'_1, Z'_2, \dots, Z'_N\}$ ,  $Z'_i \in R^{F'}$ , which can be calculated by the following equation:

$$Z'_u = \text{ReLU}\left(W \cdot \text{Concat}\left(Z_u, \vec{h}_u\right) + b\right), \quad (8)$$

where  $\vec{h}_u$  is the output vector of the aggregate attention layer, which aggregates the output of  $K$  heads of attention in the upper layer, which can be calculated by the following equation:

$$\vec{h}_u = \sigma\left(\sum_{k=1}^K \alpha_h W_h \vec{h}_u^k\right), \quad (9)$$

where  $\alpha_h$  is the attention coefficient of the aggregation layer,  $W_h$  is the weight vector of the aggregation layer, and the output of the  $k$ -head attention is aggregated through the sigmoid function to obtain the output of node  $u$ , the calculation method of  $\alpha_h$  as shown in the following equation:

$$\begin{aligned} e_h^k &= \gamma\left(\beta^T \text{Concat}\left[W_h z_u, W_h \vec{h}_u^k\right]\right) + b_h, \\ \alpha_h &= \text{softmax}_k(e_h^k) = \frac{\exp(e_h^k)}{\sum_{j=1}^K \exp(e_h^k)}, \end{aligned} \quad (10)$$

where  $e_h^k$  represents the influence of  $k$ <sub>th</sub> head attentional output  $\vec{h}_u^k$  on the node when calculating the aggregation expression of node  $u$ .  $\vec{h}_u^k$  is the output of node  $u$  through the  $k$ <sub>th</sub> head attention, and it can be calculated by (11). Vectors  $z_u$  and  $\vec{h}_u^k$  are multiplied by the Shared weight vector  $W_h$  and then concatenated as the input of the single-layer feedforward neural network;  $b_h$  is the bias.  $\beta$  is the weight vector of the neural network layer, and  $\gamma$  is the LeakyReLu nonlinear function with an input gradient of 0.2. In order to make the coefficients between different neighbor nodes easy to compare, the attention coefficient  $\alpha_h$  of each neighbor node is obtained by using the softmax function:

$$\vec{h}_u^k = \sigma\left(\alpha_{uv}^k W Z_v \mid v \in N(u)\right), \quad (11)$$

where  $\alpha_{uv}^k$  represents the attention coefficient of the first layer,  $W$  is the weight vector,  $Z_u$  and  $Z_v$  are the representation of user learned from item domain, and the calculation method of  $\alpha_{uv}^k$  is as shown in the following equation:

$$\begin{aligned} e_{uv}^k &= \gamma(\beta^T \text{Concat}[W_k z_u, W_k z_v]) + b_k, \\ \alpha_{uv}^k &= \text{softmax}_v(e_{uv}^k) = \frac{\exp(e_{uv}^k)}{\sum_{j \in N(u)} \exp(e_{uj}^k)}, \end{aligned} \quad (12)$$

where  $e_{uv}^k$  represents the influence of neighbor  $v$  on the node when calculating the expression of node  $u$ , the calculation process is consistent with  $e_h^k$ ,  $W_k$  is the shared weight vector, and  $b_k$  is the bias. We get the attention vector  $\alpha_{uv}^k$  by normalizing through sigmoid.

**4.3. Mask Mechanism.** The size of the network determines the percentage of the mask; the more extensive the network, the larger the proportion of the nodes in the mask, and there are three ways to calculate the mask node:

- (i) The probability of  $1 - \theta$  is randomly sampling the feature through mask vector from the neighbor node to represent the corresponding feature of the mask node.
- (ii) The probability of  $1 - \theta$  is randomly sampling the feature through mask vector from the neighbor node to represent the corresponding feature of the mask node. The probability of  $\theta/2$  represents the corresponding features of the masked node by randomly sampling the features from other nodes in the network.
- (iii) The probability of  $1 - \theta$  is randomly sampling the feature through mask vector from the neighbor node to represent the corresponding feature of the mask node. The probability of  $\theta/2$  does not deal with the node features of the masked node.

$Z_{imask}$  stands for mask node in item domain and  $Z_{umask}$  stands for mask node in social domain, and the calculation formula is as shown in the following equations:

$$Z_{imask} = \begin{cases} g(Z_{imask}, Z_{irandom}), & \pi = 1 - \theta, \\ Z_{random}, & \pi = \theta/2, \\ Z, & \pi = \theta/2, \end{cases} \quad (13)$$

$$Z_{umask} = \begin{cases} g(Z_{umask}, Z_{urandom}), & \pi = 1 - \theta, \\ Z_{random}, & \pi = \theta/2, \\ Z, & \pi = \theta/2, \end{cases} \quad (14)$$

$g(Z_{imask}, Z_{irandom})$  indicates that, by randomly selecting a neighbor of masked node in the item domain and transmitting the features of neighbor to masked node by mask vector, the mask vector is learned in the training process;  $Z_{random}$  indicates randomly selecting a node in the item domain and propagating the features of the node to masked node;  $Z$  indicates keeping the features of the masked node.  $f(z_{umask}, z_{urandom})$  function indicates randomly selecting a

neighbor of masked node in the social domain and transmitting the features of neighbor to masked node;  $z_{random}$  indicates randomly selecting a node in the social domain and propagating the features of the node to masked node; and  $z$  represents the original embedding of the masked node.  $\pi$  represents the strategy for processing mask nodes.

**4.4. Complexity Analysis.** The time cost of the HA-RS model is composed of two parts. One is to calculate the users' representation in the item domain. Each user needs to be calculated once, the time complexity is  $O(N)$ , and  $N$  is the number of users. The second is aggregating the features of neighbor nodes in the social domain. The time complexity of the aggregation process is  $O(NLT)$ , where  $N$  is the number of users,  $L$  is the maximum neighbor order of the aggregation, and  $T$  is the number of neighbors of each user. After the reduction strategy, the number of neighbors of each user is less than or equal to  $T$ . Since  $T$  is much less than  $N$ , and  $M$  and  $L$  have small values, the time complexity of the HA-RS model is less than  $O(N^2)$ .

## 5. Experiments

In this section, we conduct experiments to evaluate the performance of HA-RS on four datasets. Specifically, we aim to answer the following two research questions: how does HA-RS perform compared with the state-of-the-art social recommendation. How do different components (i.e., the Context-NE model, the attention model, and the mask mechanism) affect HA-RS?

### 5.1. Experimental Settings

**5.1.1. Datasets.** In this section, we conduct experiments that are including transductive learning, inductive learning, ablation experiment, and parameter selection to evaluate the performance of HA-RS on four datasets. Specifically, we aim to answer the following two research questions: how does HA-RS perform compared with the state-of-the-art social recommendation. How do different components (i.e., the Context-NE model, the attention model, and the mask mechanism) affect HA-RS?

### 5.2. Experimental Settings

**5.2.1. Datasets.** We use the citation network datasets for transductive learning, which is a collection composed of references and relationships between references. In this paper, three citation networks (Cora, CiteSeer, and PubMed) were used for experiments. The description of the dataset is shown in Table 1. In each network, nodes and edges are papers and undirected citations, respectively. The node content is constructed by extracting the words from the documents.

We use Yelp for inductive learning that is an online location-based social network. It contains business tables, reviews, tips, user information, and check-in tables. The business table lists the restaurant's name, geographic

TABLE 1: The description of the citation network dataset.

Dataset	No. of nodes	No. of edges	No. of classes	No. of features
CiteSeer	3327	4732	6	3703
Cora	2708	5429	7	1433
PubMed	19717	44338	3	500

location, opening hours, cuisine categories, average star rating, and so forth. The review table lists the restaurant's star rating, evaluation content, evaluation time, and support rate for the evaluation. In this paper, we transform the ratings that are larger than 3 as the liked items by this user and contain 141804 users and 17625 items.

**5.2.2. Baselines.** For the transductive learning task, we compare HA-RS with various state-of-the-art baselines, including three classical recommendation models: multi-layer perceptron (MLP), label propagation (LP) [8], and graph embedding (DeepWalk) [33]. Besides, we also compare our proposed model with four graph convolutional based recommendation models: graph convolution with Chebyshev filters (Chebyshev) [18], graph convolutional network (GCN) [16], graph attention networks (GAT) [19], and Masked Graph Convolutional Network (Masked GCN) [28].

- (i) MLP: It has multiple layers of neurons, which is a deep neural network, which cannot effectively process the graph data.
- (ii) LP: It uses local information from truncated random walks as input, learning a representation of vertices in a network that encodes social relations in a continuous vector space.
- (iii) Chebyshev: It provides strict control over the local support of filters to extract local and stationary features through graph convolutional layers. The method is computationally more efficient by avoiding an explicit use of the Graph Fourier basis.
- (iv) GCN: It uses an efficient layerwise propagation rule to learn hidden layer representations; the rule is based on a first-order approximation of spectral convolutions on graphs. However, it scales linearly in the number of graph edges.
- (v) GAT: It allows for implicitly assigning different importance to different nodes within a neighborhood while dealing with different sized neighborhoods and does not depend on knowing the entire graph structure upfront.
- (vi) Masked GCN: It only propagates a specific part of the node attributes via a mask vector learned for each node.

For the inductive learning task, we compare HA-RS with three classical recommendation models: BPR [34], feature enhanced latent factor model FM [35], and a state-of-the-art social recommendation model TrustSVD [24]. Besides, we also compare our proposed model with four graph

convolutional based recommendation models: GC-MC [36], PinSage [29], GraphRec [31], and DiffNet [32]. Note that GraphRec and DiffNet consider the heterogeneity of social network.

- (i) BPR: It only considered the user-item rating information for recommendation; FM and TrustSVD improve over BPR by leveraging the node features and social network information.
- (ii) GC-MC: It is one of the first few attempts that directly applied the graph convolutions for a recommendation. GC-MC defines a user-item bipartite graph from user-item interaction behavior and formulating matrix completion as a link prediction task on the bipartite graph
- (iii) PinSage: It is designed for similar item recommendations from an extensive recommender system, combining efficient random walks and graph convolutions to generate embeddings of nodes (i.e., items) that incorporate both graph structure and node feature information. However, this method homogenizes the nodes in the graph without taking into account the differences between user and user interactions and between the user and item interactions.
- (iv) GraphRec: It provides a principled approach to jointly capture interactions and opinions (rate) in the user-item graph, and the framework coherently models two graphs and heterogeneous strengths.
- (v) DiffNet: It is designed to simulate how users are influenced by the recursive social diffusion process for social recommendation. This method considers the heterogeneity of social networks, which separately learns user embedding and item embedding, finally summing the user embedding and user pleasing items embedding for prediction.
- (vi) In order to further validate the improvement obtained by HA-RS, we designed three variant models based on HA-RS.
- (vii) HA-attn has removed the hierarchical attention mechanism from the model, HA-mask has removed the mask mechanism from the model, and RS only considers the heterogeneity of the social network, while removing the hierarchical attention mechanism and the mask mechanism.

**5.2.3. Evaluation Metrics.** On the citation network dataset, the average classification accuracy (with standard deviation) was used to measure the model performance, and we reuse the metrics already reported in Masked Graph Convolutional Network for the state-of-the-art techniques.

On the Yelp dataset, as we focus on recommending top-N items for each user, we use two widely adopted ranking based metrics: hit ratio (HR) and normalized discounted cumulative gain (NDCG). For these two indicators, the larger the value, the better the performance. We reuse the metrics already reported in A Neural Influence Diffusion

Model for Social Recommendation for the state-of-the-art techniques.

HR measures the number of items that the user likes in the test data that has been successfully predicted in the top-N ranking list, and its calculation formula is shown in the following equation:

$$\text{HR}@N = \frac{\sum_{u \in N(u)} \text{HR}@N_u}{T}. \quad (15)$$

The denominator is all the test sets, and the numerator is the sum of the number of each user's top- $N$  recommendation list in test sets.

NDCG considers the hit positions of the items and gives a higher score if the hit items are in the top positions and its calculation formula is shown in the following equation:

$$\text{NDCG} = \frac{1}{\log_2(1 + \text{rank}_{\text{pos}})}, \quad (16)$$

where  $\text{rank}_{\text{pos}}$  indicates the rank of the positive item (successful prediction item).

**5.3. Parameter Setting.** During training, when learning user representation in the item domain, we apply the grid search to set the hyper-parameters  $M$ ,  $C$ , and  $L_w$ . By using the Masked LM model in Bert [37] for reference, this paper sets  $\theta$  to 20%. We aggregate the feature embedding of  $L$ -layer neighbor in the social domain and set layer  $L=2$ ; the user's maximum number of neighbors  $T$  is set to 15 based on the grid search results. L2 regularization was used to prevent overfitting. The attenuation coefficient is set to 0.00005 when the dataset was the citation network and 0.001 when the dataset was Yelp. User nodes in Yelp dataset output embedded dimension  $D$  after passing through the Context-NE model. In the first layer of attention and the aggregation layer of attention, the output of each layer is also set to  $D$  dimension. The first attentional mechanism is composed of  $K=3$  attentional heads, whose output is the input of the second attentional head. We used Adam as the optimization method for all the models that relied on gradient descent-based methods, with an initial learning rate of 0.001.

#### 5.4. Experimental Results

**5.4.1. Accuracy of Transductive Learning on Citation Network.** For the transductive learning, the results of classification accuracy on the three datasets are shown in Table 2. It can be seen from Table 2 that the performance of our model is better than other baselines. More specifically, the accuracy of the model in the three datasets of the citation network in this paper is 1.9%, 2%, and 2.1% higher than the GAT model, respectively, indicating that the HA-RS with mask mechanism and hierarchical attention can improve the performance.

Masked GCN propagates partial attributes instead of the entire ones via a mask vector learned for each node, and the Masked GCN significantly improves the performances compared to GCN and GAT. However, according to the

TABLE 2: Summary of results in terms of classification accuracies, for Cora, CiteSeer, and PubMed.

Methods	Cora (%)	CiteSeer (%)	PubMed (%)
MLP	55.1	46.5	71.4
LP	68.0	45.3	63.0
DeepWalk	67.2	43.2	65.3
Chebyshev	81.2	69.8	74.4
GCN	81.5	70.3	79.0
GAT	83.0	72.5	79.0
Masked GCN	84.4	73.8	80.2
HA-RS	84.9	74.5	81.1

paper of the Masked Graph Convolutional Network, the running time of Masked GCN is 1.24 times compared to that of GAT on average, and the extra time is mainly utilized to learn the parameters of masks. HA-RS only learns mask parameters of a few nodes; it can save much time. Besides, our model HA-RS outperforms Masked GCN.

**5.4.2. Experimental Results of Inductive Learning on Yelp.** For the inductive learning, the results in terms of HR and NDCG are shown in Tables 3 and 4. Table 3 shows the HR@10 and NDCG@10 results with varying output dimension size  $D$ , and we have the following observations:

- (i) On both HR and NDCG, the graph convolutional based recommendation models outperform classical recommendation models. Graph convolution methods considering network heterogeneity, such as GraphRec and DiffNet, have better performance than GC-MC and PinSage. As DiffNet is the best baselines, our model HA-RS improves over DiffNet range of 3% to 3.5% on HR as the output dimension size  $D$  increases from 16 to 64 and range of 1.2% to 5.3% on NDCG. It indicates that this paper divides the heterogeneous social network into the item domain and social domain, and the user representation obtained is more suitable for a recommendation.
- (ii) We find that the performance of all models does not increase as the output latent dimension size  $D$  increases from 16 to 64. The BPR and FM models achieve the best performance at  $D=32$ , and other models achieve optimal performance at  $D=64$ . It might be concluded that BPR only considered the user-item rating information for the recommendation; a too large dimension increases the noise in the representation, resulting in a decrease in the recommended performance. Although FM leveraging the node features and social network information, the feature extraction ability of the model is too weak.

Table 4 shows the HR@ $N$  and NDCG@ $N$  results with varying top-N recommendation size  $N$ . From the results, we also find similar observations as in Table 3, with our proposed model HA-RS always showing the best performance. Our model HA-RS improves over DiffNet range of 2.3% to 4.0% on HR as the top-N recommendation size  $N$  increases

TABLE 3: HR@10 and NDCG@10 comparisons for different dimension sizes D.

Models	Yelp					
	HR			NDCG		
	D = 16	D = 32	D = 64	D = 16	D = 32	D = 64
BPR	0.2443	0.2632	0.2617	0.1471	0.1575	0.1550
FM	0.2768	0.2835	0.2825	0.1698	0.1720	0.1717
TrustSVD	0.2853	0.2880	0.2915	0.1704	0.1723	0.1738
GC-MC	0.2875	0.2902	0.2937	0.1657	0.1686	0.1740
PinSage	0.2952	0.2958	0.3065	0.1758	0.1779	0.1868
DiffNet	0.3366	0.3437	0.3477	0.2052	0.2095	0.2121
HA-RS	0.3487	0.3542	0.3596	0.2122	0.2179	0.2235

TABLE 4: HR@N and NDCG@N comparisons for different top-N values.

Models	Yelp					
	HR			NDCG		
	N = 5	N = 10	N = 15	N = 5	N = 10	N = 15
BPR	0.1713	0.2632	0.3289	0.1243	0.1575	0.1773
FM	0.1881	0.2835	0.3463	0.1359	0.1720	0.1895
TrustSVD	0.1906	0.2915	0.3693	0.1385	0.1738	0.1983
GC-MC	0.1932	0.2937	0.3652	0.1420	0.1740	0.1922
PinSage	0.2099	0.3065	0.3873	0.1536	0.1868	0.2130
DiffNet	0.2276	0.3477	0.4232	0.1679	0.2121	0.2331
HA-RS	0.2357	0.3557	0.4404	0.1803	0.2185	0.2451

from 5 to 15 and range of 2.9% to 5.1% on NDCG. Based on the experiment results, we could empirically conclude that our proposed HA-RS model outperforms all the baselines under different output dimensions and different recommendation size  $N$ .

#### 5.4.3. Ablation Experiment

(1) *Ablation Experiment and Cold-Start Analysis.* We set the ablation experiment in Yelp dataset to verify the effectiveness of the model, and we set the dimension size as 64 and the  $N$  as 10 in top-N recommendations. The experimental results are shown in Figure 3.

According to Figures 3(a) and 3(b), we observe that HA-RS outperforms the others on HR and NDCG. When removing the hierarchical attention mechanism, the recommendation performance in HR and NDCG decreased slightly, 4.8% and 3.8%, respectively, but there is a great decline in performance of HR and NDCG when removing the mask mechanism, 10.2% and 9%, respectively. It indicates that both the reduced hierarchical attention mechanism and the mask mechanism will affect the performance of the model, and removing the mask mechanism has a more significant impact on the model.

Besides, removing both the hierarchical attention mechanism and the mask mechanism, the performance of the RS was the worst; however, it also outperforms PinSage that uses both social network information and node feature information. The main reason is that PinSage processes all

nodes in the full graph in the same way and does not explicitly distinguish between the user (board) and item (pins) nodes which does not consider the heterogeneity of the social network. It demonstrates the superiority of our model by dividing the heterogeneous social graph into two domains.

We verified the cold-start problem on the Cora dataset, set  $N_c$  of the cold-start users as 0, 50, 100, and 150, respectively, and compared the model HA-RS with its three variants. We regard the cold-start users as the masked nodes and learned by mask vector in HA-RS and HA-attn while, in HA-mask and RS, the cold-start user is randomly initialized; the experimental results are shown in Table 5.

For cold-start users, there is very little information available about a recommendation, and the recommendation result obtained by randomly initializing user embedding is often not ideal. In the method based on social networks, the cold-start problem is alleviated by combining the information of friends. In this paper, cold-start users are regarded as masking nodes, and user embedding is learned by the mask vector. From Table 5, we can conclude that the HA-RS and HA-attn (with mask mechanism) outperform HA-mask and RS (without mask mechanism). Therefore, it can say that the mask mechanism can solve the cold-start problem.

(2) *Parameter Selection.* In this section, we mainly explore the impact of different parameters of HA-RS. We study the impact of L-layer neighbor in the aggregating process in the social domain, the effect of a different number of user neighbors, and the effect of different mask proportions.

We vary the layer of neighbor from 1 to 3 and set the number of user neighbors in every layer from 5 to 20; the performance of HR and NDCG is shown in Figure 4. We can observe from Figure 4 that HA-RS has the best performance when the number of user neighbors is 15; the performance of HA-RS has declined dramatically when the number of neighbors increases to 20. Meanwhile, we can find that aggregating 2-layer neighbor in HA-RS outperforms aggregating 1-layer or 3-layer neighbor; this is because 1-layer cannot capture the higher-order relationships between users in the social domain. Nevertheless, 3-layer may bring massive noise to the model. The above observation has proved that aggregating the features of too many neighbors will increase the noise in the user's representation, which is described in Section 4.2. Other related studies have empirically found similar trends, with the best layer size set as  $L = 2$  [32, 38].

Consider that Masked GCN propagates partial attributes instead of the entire ones via a mask vector learned for each node, and we will discuss that the different mask ratios influence the performance of the model. We apply the mask mechanism of this paper to the original GCN and GAT models, respectively, expressed as GCN-mask and GAT-mask, and compare with HA-RS proposed in this paper.

As mentioned above, the mask mechanism plays a vital role in improving the model. Six groups of different mask proportions are set in the experiment. When the proportion is set to 0%, it is the GCN, GAT, and HA-RS without mask mechanism, and the maximum mask proportion is set to 10%. The accuracy of the three methods in the citation network is shown in Figure 5.

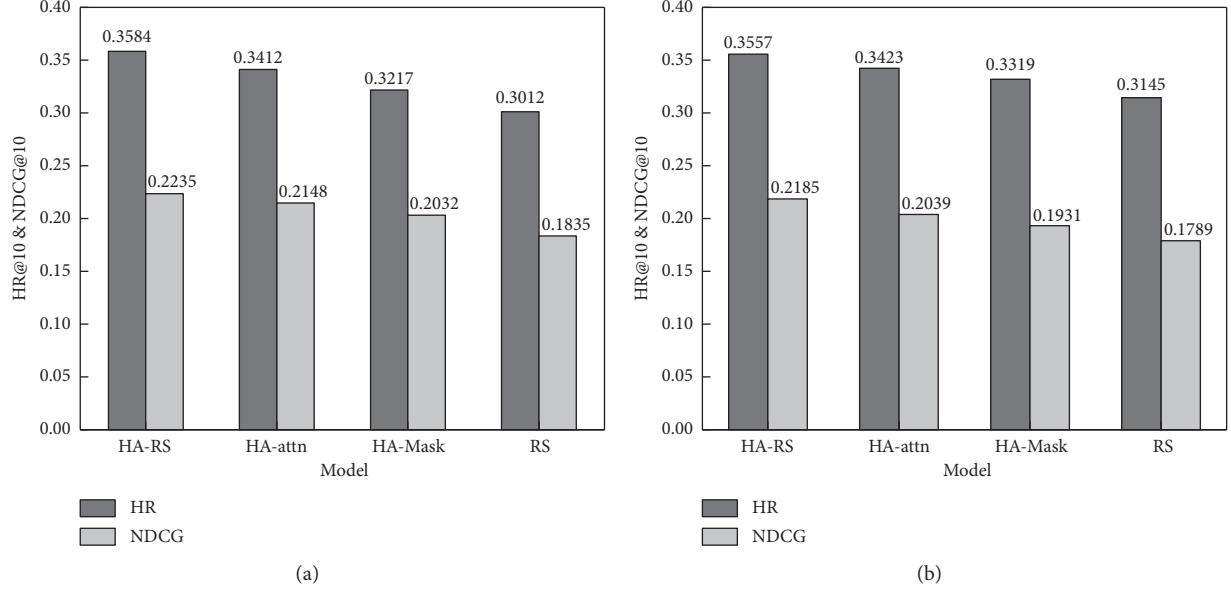


FIGURE 3: The results of the ablation experiment. (a) HR@10 and NDCG@10 comparisons for dimension size = 64. (b) HR@10 and NDCG@10 comparisons for the top-10 values.

TABLE 5: Results of different cold-start users on Cora dataset.

Model	$N_c = 0$ (%)	$N_c = 50$ (%)	$N_c = 100$ (%)	$N_c = 150$ (%)
RS	83.4	83.1	82.6	82.1
HA-attn	84.5	84.3	84.0	84.6
HA-mask	83.7	83.4	83.0	82.5
HA-RS	84.9	84.6	84.3	83.8

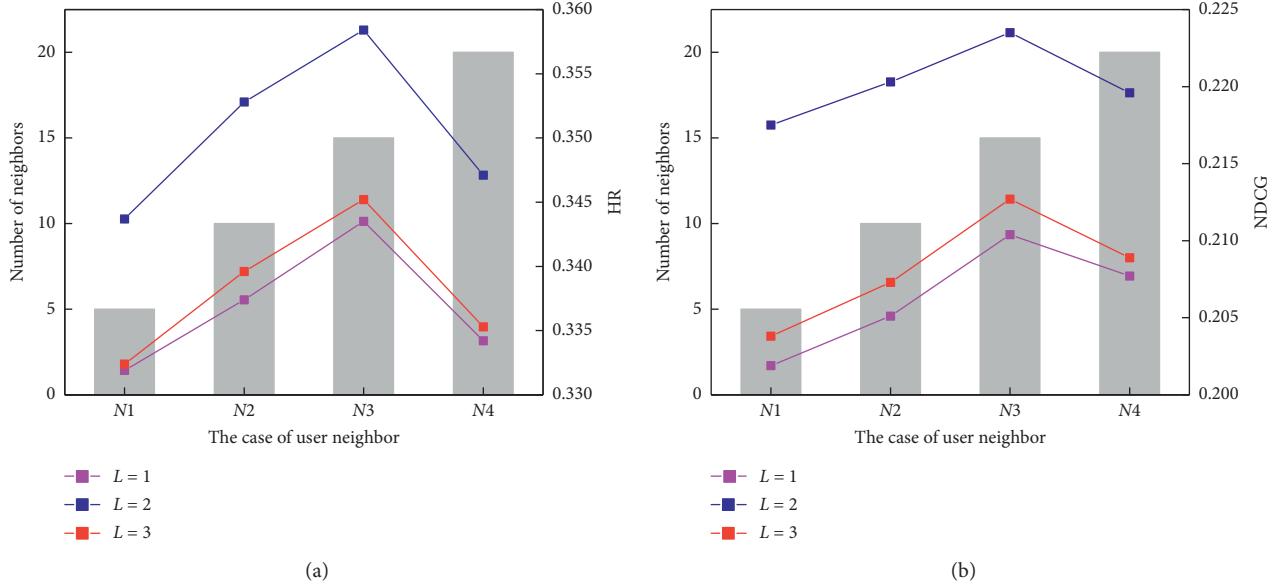


FIGURE 4: The background histograms indicate the four cases of user neighbor; meanwhile, the lines demonstrate the performance of HR/ NDCG with different layers. (a) Performance of HR. (b) Performance of NDCG.

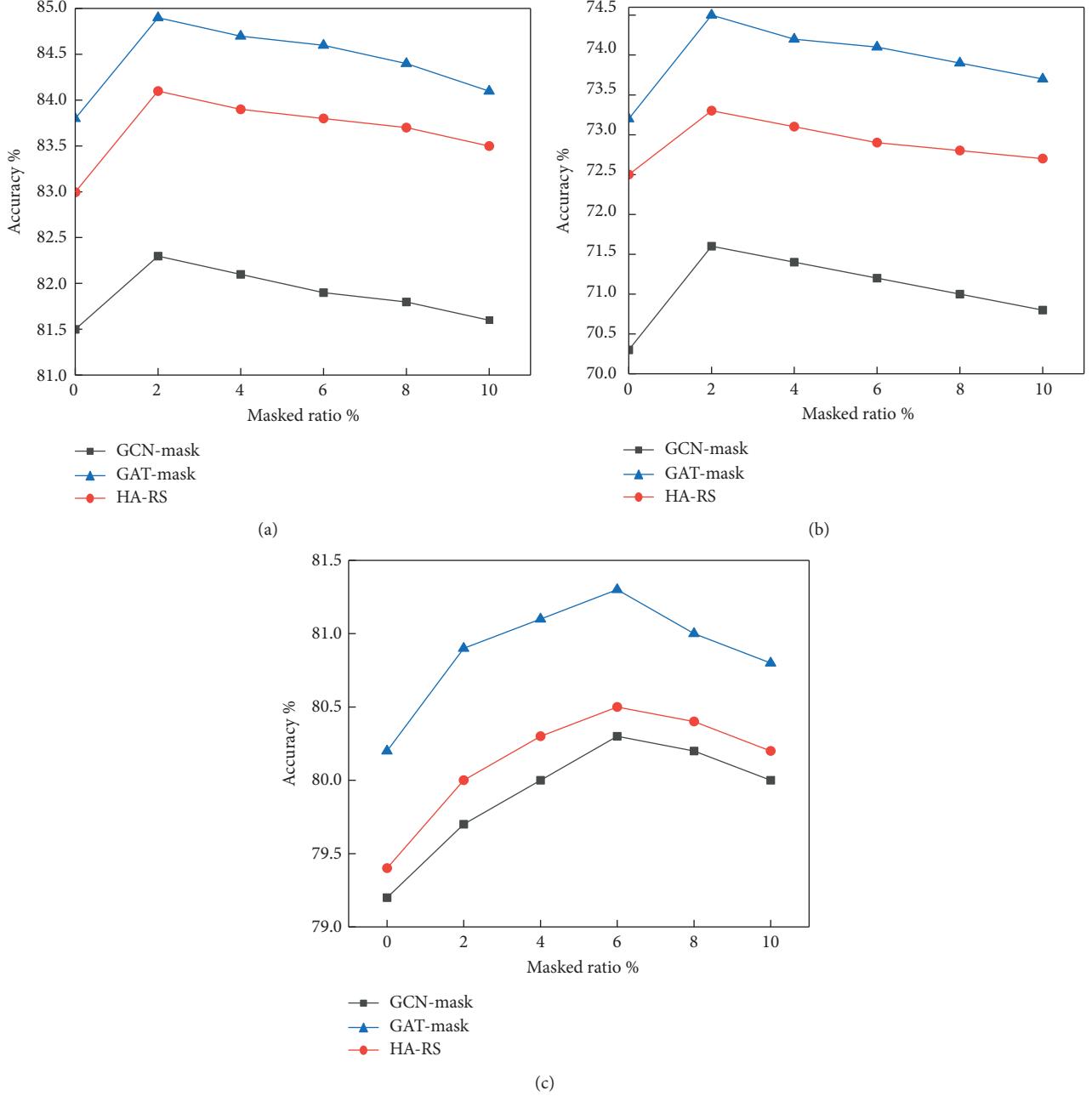


FIGURE 5: Results of different masked ratios on the citation dataset. (a) Cora dataset. (b) CiteSeer dataset. (c) PubMed dataset.

According to Figure 5, we can find that the HA-attn outperforms GCN and GAT when the mask rate is 0%, and it is also proved that dividing the heterogeneous network into the social domain and item domain to learn user representation is better for classification and recommendation. In the dataset of Cora and CiteSeer, the accuracy of the three models reached the highest when the mask proportion was 2%, while the accuracy of the models decreased with the increase of the mask proportion. On PubMed dataset, the mask ratio reached the highest value when it was 6% and then decreased slowly. The main reason is that there are more nodes in PubMed dataset, 6 times more than in Cora and CiteSeer. At a mask ratio of 10% in three datasets, the

three methods perform better than those without a masking mechanism. This shows that the mask mechanism can indeed affect the performance of the model, and the proportion of the mask is related to the network scale. The proportion of the mask that achieves the best performance increases with the increase of the network scale.

## 6. Conclusion

In this work, we explore the problem of extracting more abundant features of user representation for the recommendation in heterogeneous social networks. Considering that the behavior of users in the item domain and the social

domain is inconsistent, we learn that user representation in the HA-RS is divided into two parts. First, we learn the user representation with user behavior in the item domain by the Context-NE model and then through the hierarchical graph attention model to learn user embedding with social impact in the social domain. The mask mechanism in the model also improves the flexibility of the network. As new nodes are added, we can regard it as a masked node. A large number of experiments on real datasets show that our model performs better than the existing techniques above. Future research directions will focus on how to use the geographical location and timing information better to improve the quality of the algorithm further.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The work was partially supported by the National Natural Science Foundation of China under Grant nos. 81901389 and 71701026; the China Postdoctoral Science Foundation under Grant no. 2019M653400; the Sichuan Science and Technology Program under Grant nos. 2019YFS0236 and 2018GZDZX0039; the Key Project for Soft Science of Meteorology under Grant no. 201704; and the Youth Foundation for Humanities and Social Sciences of Ministry of Education of China under Grant no. 17YJC190035.

## References

- [1] S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, “Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure,” in *Proceedings of the 2012 International Conference on Computer Science and Service System*, pp. 2038–2041, IEEE, Nanjing, China, August 2012.
- [2] H. Huo, Z. Wei, L. Liang, and L. Yang, “Collaborative filtering recommendation model based on convolutional denoising auto encoder,” in *Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing*, Kunming, China, August 2017.
- [3] X. Xiong, S. Qiao, Y. Li et al., “ADPDF: a hybrid attribute discrimination method for psychometric data with fuzziness,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 265–278, 2019.
- [4] X. Xiong, C. Xie, R. Zhao, Y. Li, S. Ju, and M. Jin, “A clickthrough rate prediction algorithm based on users’ behaviors,” *IEEE Access*, vol. 7, pp. 174782–174792, 2019.
- [5] X. Xiong, Y. Li, R. Zhang, Z. Bu, G. Li, and S. Ju, “DGI: recognition of textual entailment via dynamic gate matching,” *Knowledge-Based Systems*, vol. 194, p. 105544, 2020.
- [6] X. Xiong, S. Qiao, Y. Li, N. Han, F. Xiong, and L. He, “Affective impression: sentiment-awareness poi suggestion via embedding in heterogeneous lbsns,” *IEEE Transactions on Affective Computing*, p. 1, 2020.
- [7] X. Xiong, F. Xiong, J. Zhao, S. Qiao, Y. Li, and Y. Zhao, “Dynamic discovery of favorite locations in spatio-temporal social networks,” *Information Processing & Management*, vol. 57, no. 6, Article ID 102337, 2020.
- [8] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using Gaussian fields and harmonic functions,” in *Proceedings of the Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, Washington, DC, USA, August 2003.
- [9] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. S. Olkof, “Learning with local and global consistency,” *Advances in Neural Information Processing Systems*, vol. 16, no. 3, 2004.
- [10] P. De Meo, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, “Providing recommendations in social networks by integrating local and global reputation,” *Information Systems*, vol. 78, pp. 58–67, 2018.
- [11] X. Xiong, S. Qiao, N. Han et al., “Where to go: an effective point-of-interest recommendation framework for heterogeneous social networks,” *Neurocomputing*, vol. 373, pp. 56–69, 2020.
- [12] X. Xiong, S. Qiao, Y. Li, N. Han, G. Yuan, and Y. Zhang, “A point-of-interest suggestion algorithm in multi-source geo-social networks,” *Engineering Applications of Artificial Intelligence*, vol. 88, Article ID 103374, 2020.
- [13] J. Tang, S. Wang, X. Hu et al., “Recommendation with social dimensions,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, February 2016.
- [14] A. M. Elkahky, Y. Song, and X. He, “A multi-view deep learning approach for cross domain user modeling in recommendation systems,” in *Proceedings of the 24th International Conference on World Wide Web*, pp. 278–288, Florence, Italy, May 2015.
- [15] C. Wang, M. Niepert, and H. Li, “Recsys-dan: discriminative adversarial networks for cross-domain recommender systems,” *IEEE Transactions on Neural Networks and Learning Systems*, 2019.
- [16] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April 2017.
- [17] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: methods and applications,” *IEEE Data Engineering Bulletin*, vol. 40, no. 3, pp. 52–74, 2017.
- [18] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” pp. 3844–3852, 2016, <http://arxiv.org/abs/1606.09375>.
- [19] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April–May 2018.
- [20] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The Adaptive Web, Methods and Strategies of Web Personalization, Vol. 4321 of Lecture Notes in Computer Science*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., pp. 325–341, Springer, Berlin, Germany, 2007.
- [21] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” *Communications of The ACM*, vol. 35, no. 12, pp. 61–70, 1992.
- [22] F. Xiong, X. Wang, S. Pan, H. Yang, H. Wang, and C. Zhang, “Social recommendation with evolutionary opinion dynamics,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

- [23] B. Yang, Y. Lei, J. Liu, and W. Li, “Social collaborative filtering by trust,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.
- [24] G. Guo, J. Zhang, and N. Yorke-Smith, “Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, TX, USA, January 2015.
- [25] Z. Li, F. Xiong, X. Wang, H. Chen, and X. Xiong, “Topological influence-aware recommendation on social networks,” *Complexity*, vol. 2019, Article ID 6325654, 2019.
- [26] F. Xiong, W. Shen, H. Chen, S. Pan, X. Wang, and Z. Yan, “Exploiting implicit influence from information propagation for social recommendation,” *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.
- [27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [28] L. Yang, F. Wu, Y. Wang, J. Gu, and Y. Guo, “Masked graph convolutional network,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, S. Kraus, Ed., pp. 4070–4077, Macao, China, August 2019.
- [29] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*, Y. Guo and F. Farooq, Eds., pp. 974–983, ACM, London, UK, August 2018.
- [30] J. Zhang, X. Shi, S. Zhao, and I. King, “STAR-GCN: stacked and reconstructed graph convolutional networks for recommender systems,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, S. Kraus, Ed., pp. 4264–4270pp. 4264–, Macao, China, August 2019.
- [31] W. Fan, Y. Ma, Q. Li et al., “Graph neural networks for social recommendation,” in *Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 417–426, Portland, OR, USA, November 2019.
- [32] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, “A neural influence diffusion model for social recommendation,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 235–244, Paris, France, July 2019.
- [33] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, New York, NY, USA, August 2014.
- [34] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: bayesian personalized ranking from implicit feedback,” 2012, <http://arxiv.org/abs/1205.2618>.
- [35] S. Rendle, “Factorization machines,” in *Proceedings of the 2010 IEEE International Conference on Data Mining*, pp. 995–1000, IEEE, Sydney, Australia, December 2010.
- [36] R. van den Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” 2017, <http://arxiv.org/abs/1706.02263>.
- [37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: pre-training of deep bidirectional transformers for language understanding,” 2018, <http://arxiv.org/abs/1810.04805>.
- [38] L. Hu, C. Li, C. Shi, C. Yang, and C. Shao, “Graph neural news recommendation with long-term and short-term interest modeling,” *Information Processing & Management*, vol. 57, no. 2, Article ID 102142, 2020.