Graph Attention Network with High-Order Neighbor Information Propagation for Social Recommendation

Fei Xiong[1] , **Haoran Sun**[1] , **Guixun Luo**[2*] , **Shirui Pan**[3] , **Meikang Qiu**[4] and **Liang Wang**[5]
[1]School of Electronic and Information Engineering, Beijing Jiaotong University
[2]School of Computer Science and Technology, Beijing Jiaotong University
[3]School of Information and Communication Technology, Griffith University
[4]School of Computer and Cyber Sciences, Augusta University
[5]School of Computer Science, Northwestern Polytechnical University
{xiongf, haoran.sun, gxluo}@bjtu.edu.cn, s.pan@griffith.edu.au, qiumeikang@yahoo.com, liangwang0123@gmail.com

Abstract

In recommender systems, graph neural networks (GNN) can integrate interactions between users and items with their attributes, which makes GNN-based methods more powerful. However, directly stacking multiple layers in a graph neural network can easily lead to over-smoothing, hence recommendation systems based on graph neural networks typically underutilize higher-order neighborhoods in their learning. Although some heterogeneous graph random walk methods based on meta-paths can achieve higher-order aggregation, the focus is predominantly on the nodes at the ends of the paths. Moreover, these methods require manually defined meta-paths, which limits the model's expressiveness and flexibility. Furthermore, path encoding in graph neural networks usually focuses only on the sequence leading to the target node. However, real-world interactions often do not follow this strict sequence, limiting the predictive performance of sequence-based network models. These problems prevent GNN-based methods from being fully effective. We propose a **G**raph **A**ttention network with **I**nformation **P**ropagation path aggregation for **S**ocial **Rec**ommendation (GAIPSRec). Firstly, we propose a universal heterogeneous graph sampling framework that does not require manually defining meta-paths for path sampling, thereby offering greater flexibility. Moreover, our method takes into account all nodes on the aggregation path and is capable of learning information from higher-order neighbors without leading to over-smoothing. Finally, our method utilizes a gate mechanism to fuse sequential and non-sequential dependence in encoding path instances, allowing a more holistic view of the data. Extensive experiments on real-world datasets show that our proposed GAIPSRec improves the performance significantly and outperforms state-of-the-art methods.

*Corresponding author.

1 Introduction

In the era of information explosion, recommender systems are taking on increasingly important responsibilities on filtering information and make personalized recommendation for users. Collaborative filtering (CF), exemplified by matrix factorization (MF) [Koren *et al.*, 2009] and neural matrix factorization (NeuMF) [He *et al.*, 2017], is a widely used approach to learn user and item representations from historical interactions for personalized recommendation. The predictive advantages of incorporating user's historical interactions are demonstrated by singular value decomposition++ (SVD++) [Koren, 2008], while neural attentive item similarity (NAIS) [He *et al.*, 2018] leverages attention networks to improve predictive accuracy. However, these approaches primarily model shallow interactions, limited to one-hop neighboring connections between users and items. GNNs offer the capacity to uncover intricate interactions between users, items, and attributes. Models like neural graph collaborative filtering (NGCF) [Wang *et al.*, 2019a], the simplified and powered graph convolution network (LightGCN) [He *et al.*, 2020] and neighborhood-enriched contrastive learning (NCL) [Lin *et al.*, 2022] address limitations of conventional methods. NGCF highlights the significance of high-order connectivity. LightGCN simplifies the approach by eliminating feature transformations and nonlinear activation modules. NCL enhances graph-based CF through contrastive learning.

Although these methods make advances in recommender systems, they are often assumed to have homogeneous graph structures and struggle with structural and semantic intricacies in heterogeneous graphs. The limitation hinders their ability to effectively model graph structures containing diverse characteristics. In heterogeneous graphs , existing methods [Tang *et al.*, 2016; Hamilton *et al.*, 2017; Liu *et al.*, 2018; Fan *et al.*, 2019a] directly stack multiple GNN layers and encounter the challenge of over-smoothing. The abundance of GNN paths prevents the recommender system from effectively capturing high-order neighbor relationships. Heterogeneous graph embedding methods based on the meta-path concept, such as HERec [Shi *et al.*, 2018] and HAN [Wang *et al.*, 2019b], could address the utilization issue of higher-order neighbors. However, these methods are

dependent on meta-path-based random walks for processing heterogeneous graphs, necessitating manually defined meta-paths, which restricts the model's expressiveness and flexibility. Additionally, these models consider only the nodes at both ends of a meta-path, ignoring all intermediate nodes, which leads to a loss of information. Furthermore, in the process of path encoding, these methods follow the path leading to the target node, but actual interactions may not always adhere to this direction. Such a discrepancy can result in inaccurate predictions, failing to accurately reflect the potential diffusion network structure.

To overcome these limitations, we introduce an innovative approach called GAIPSRec. GAIPSRec adopts the universal heterogeneous graph sampling framework we proposed, which eliminates the need for manually defining meta-paths. Additionally, it can aggregate high-order information without leading to over-smoothing. This method is capable of considering all nodes on higher-order neighbor relationships, not just the terminal ones. We employ a gate mechanism to fuse sequential (LSTM) and non-sequential(coverage mechanism) dependence, encoding the information in each node along the sampled path. The introduction of the coverage mechanism improves the utilization coverage of all representation information and enhances the memory properties of the attention mechanism, thus correcting erroneous attention allocation. By incorporating the cross-dependence, our approach gains a more holistic view of the data and reflects the potential diffusion network structure. GAIPSRec is able to capture the structural and semantic information of the heterogeneous graphs by considering interaction neighbor nodes and social neighbor nodes along the sampled paths. Following path encoding, GAIPSRec performs two-level path aggregation using attention mechanism, merging latent vectors from multiple sampled paths into the final node embedding. By integrating multiple sampled paths, our model adeptly learns comprehensive semantics embedded within the heterogeneous graphs.

In summary, this research brings several significant contributions:

- We propose a universal heterogeneous graph sampling framework, which does not require manually defined meta-paths for the path sampling process, thereby enhancing the model's flexibility. It effectively captures high-order neighbor information without leading to over-smoothing.
- We consider the information from all neighboring nodes along the entire path during path encoding, in contrast to the meta-path sampling method that focuses solely on the terminal nodes.
- We employ a gate mechanism to fuse sequential and non-sequential dependence, allowing us to view interaction data from a more comprehensive perspective.
- We carry out empirical studies using two datasets sourced from real-world scenarios. The outcomes of these experiments demonstrate a clear superiority of GAIPSRec over existing state-of-the-art baseline models.

2 Related Work

Graph Neural Network. GNNs initially emerge in the spectral domain due to their computational complexity [Bruna *et al.*, 2013]. Defferrard et al introduced Chebyshev polynomials [Defferrard *et al.*, 2016] to reduce computational complexity. GCN is derived from spectral methods but has a spatial interpretation due to the specific approximations used, and this approach has had a significant impact on the field of research in spatial GNNs. [Kipf and Welling, 2016]. With the application of the attention mechanism in deep learning models [Vaswani *et al.*, 2017], graph attention networks (GAT) uses it for aggregating information from neighboring nodes [Veličković *et al.*, 2017]. Furthermore, an algorithm known as FastGCN [Chen *et al.*, 2018] introduces layer-wise sampling to reduce variance. Cluster-GCN, an efficient algorithm for training deep and large GCNs, uses subgraph sampling to increase parallelism [Chiang *et al.*, 2019]. Despite their efficiency, these methods solely rely on user-item interaction data, and suffer from data sparsity. Adopting social recommendation can effectively alleviate this issue [Fan *et al.*, 2018; Huang *et al.*, 2021].

Social Recommendation. Early heterogeneous graph-based recommendation methods employ first-order social neighbors [Tang *et al.*, 2013]. Both matrix factorization (SocialMF) [Jamali and Ester, 2010] and probabilistic matrix factorization (SoRec) [Ma *et al.*, 2008] for recommendation in social rating networks model social influence as a regularization term. In contrast, social recommendation models based on GNNs leverage social neighbors and multiple types of connections [Fan *et al.*, 2019a]. An influence diffusion neural network based model(Diffnet) [Wu *et al.*, 2019a] and another neural influence and interest diffusion network based model (Diffnet++) [Wu *et al.*, 2020] use attention mechanism and k-order information diffusion to make social recommendation. Some methods further utilize user communities to enrich recommender systems [Liu *et al.*, 2020; Xing *et al.*, 2022; Fatemi and Tokarchuk, 2013].

The above methods tend to combine representations of users in the social graph with the user-item graph and assign appropriate weights. However, directly stacking multiple layers of GNNs can result in over-smoothing and limit the learning potential of higher-order neighbor information in GNN-based recommendation models. Even when involving high-order neighbors, it necessitates the definition of meta-paths and focuses only on the nodes at the ends. Our proposed model, GAIPSRec, is designed to handle them effectively.

3 Proposed Method

The GAIPSRec framework, as illustrated in Fig. 1, firstly random walks with restart based on length are performed on the heterogeneous recommendation graph to obtain sampled paths and path neighbors (starting and ending nodes). Various path encoders, such as LSTM, GRU, mean aggregation, and coverage mechanism with sequential dependence, are employed to model the information propagation process along the sampled paths. The GAT is then utilized to construct subgraphs by aggregating paths. Finally, attention mechanism is applied to aggregate subgraph representations of different

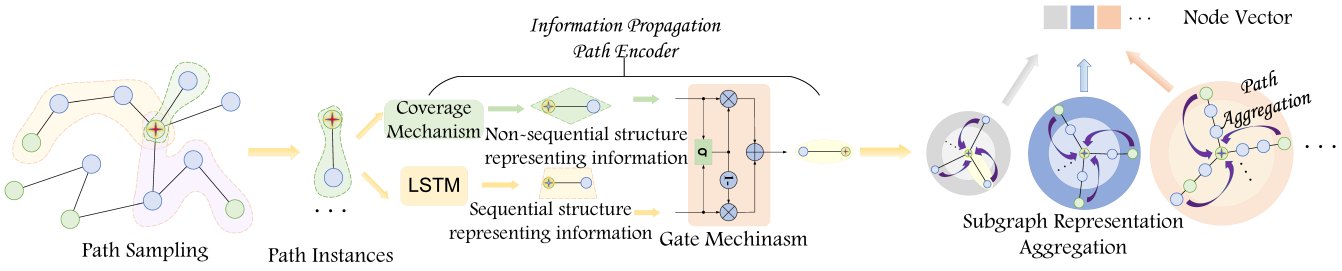


Figure 1: Method framework.

lengths, resulting in user embedding and item embedding. The embedding is further utilized to perform rating prediction tasks.

3.1 Path Sampling for Information Propagation

In this work, we consider users and items as nodes during the sampling stage for recommendation in heterogeneous graphs containing both social and rating relationships. We use Random Walk with Restart (RWR) to simulate the paths of information propagation in the recommendation network and sample a set number of paths of varying lengths for each type of node. By employing this sampling strategy, we sample a subset of paths from the heterogeneous graph with the combination of user-item interactions and user-user relationships, and this sampling strategy significantly reduce the number of paths. This enables our method to consider higher-order neighbor relationships and mitigate the risk of over-smoothing in multiple aggregations of GNNs. At each step of the random walk, there is a certain probability of restarting and returning to the start node, which increases the influence of the start node on the walk sequence. Assuming the walk starts from a user node, at each step, the next node is selected based on the probability distribution.

$$p = \begin{cases} p_{rwr}, & \text{restart;} \\ \frac{1-p_{rwr}}{|N_{neigh}|}, & \text{select any neighbor,} \end{cases} \tag{1}$$

where p_{rwr} denotes the probability of the random walk returning to the start node, and $|N_{neigh}|$ denotes the number of first-order neighbors. The probability for selecting neighbors is set to follow uniform distribution. We denote the probability of a user node selecting the next node as follows:

$$p_u = \begin{cases} \frac{|N_{item}|}{|N_{neigh}|}, & \text{select an item(rating relationships);} \\ \frac{|N_{user}|}{|N_{neigh}|}, & \text{choose a user(social relationships),} \end{cases} \tag{2}$$

where $|N_{item}|$ denotes the number of ratings given by the user, and $|N_{user}|$ denotes the number of social connections of the user. The next choice for an item node is only the rating relationships. By setting the above sampling probability, the model can sample the paths that are most closely related to the central node. Compared to random walk that only considers first-order neighbor nodes, random walk with restart sampling can capture more global graph structural information. The restart probability, p_{rwr}, is a hyperparameter of the model. When p_{rwr} is small, the sampling closely resembles random walk that only considers neighbor nodes, while a large p_{rwr} may result in insufficient sampling of the specified paths. In this part, paths with different lengths are sampled for each type of nodes, enabling the learning of information propagation and neighbor influence at different orders. Finally, attention mechanism is used to aggregate the representations of subgraphs with different lengths to update the node representations.

3.2 Information Propagation Path Encoder

Previous graph-based social recommendation models often separately train user embeddings in the recommendation network and social network before fusion [Fan *et al.*, 2019a; Wu *et al.*, 2019b]. In this part, we incorporate simulated information propagation paths through sampling. This method effectively preserves the heterogeneous interaction information between users and items and allows item embedding to be influenced by social relationships. By modeling information propagation along the paths, the model can learn higher-order neighbor information. To avoid over-smoothing of node embeddings, caused by stacking multiple layers of graph neural networks, we combine the modeling of information propagation with the path aggregation. We use multiple encoders to simulate the information propagation process along the random walk paths and obtain path representations.

$$\boldsymbol{M}_{p_{u\to v}} = PE(\boldsymbol{e}_n, \boldsymbol{e}_r, \{\forall n \in N^{p_{u\to v}}, \forall r \in E^{p_{u\to v}}\}), \tag{3}$$

where $p_{u\to v}$ denotes a path instance from node u to node v, and nodes u and v are referred to as path neighbors in this paper. $\boldsymbol{M}_{p_{u\to v}}$ denotes the representation of the path information from node u to node v, and PE refers to the propagation encoder. $\boldsymbol{e}_n$ and $\boldsymbol{e}_r$ represent the embeddings of nodes and edges, respectively. Nodes include both user nodes and item nodes, while edges encompass both user-item interactions and social relationships.

The mean encoder computes the mean of the node embeddings along the path and uses it as the path representation. LSTM and GRU are used to encode the sequence of path nodes. The node ordering is set from far to near in RNN encoders, indicating that higher-order neighbors propagate information to the central node through intermediate nodes step by step. The GAIPSRec models with the mean, LSTM, and GRU encoders are denoted as GAIPSRec-mean GAIPSRec-lstm, GAIPSRec-gru, respectively.

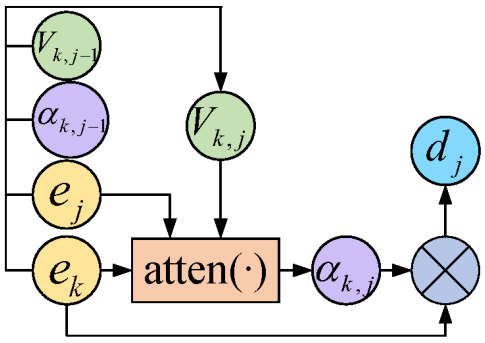


Figure 2: Diffusion-dependence attention layer with coverage.

Coverage mechanism and its fusion with sequential dependence. The actual process of information diffusion may not strictly adhere to the sequential assumption, thereby limiting the predictive performance of sequential models such as LSTM and GRU. To overcome this limitation, inspired by the coverage model in linguistics, we propose a diffusion-dependent attention layer with coverage mechanism. This mechanism enables the acquisition of non-sequential structured path representations while capturing diffusion dependence. It enhances the coverage of all representation information. Subsequently, we employ a gate mechanism to integrate non-sequential and sequential structured information of path representations, yielding the final path encoding.

Given a node $n_j \in \{n_0, n_1, \ldots, n_{N-1}\}$, whose original representation is a one-hot vector of the user/item ID, denoted as $\boldsymbol{x}_j \in \mathcal{R}^N$, where N denotes the total number of users and items. To extract more expressive and efficient node features, the original node representation $\boldsymbol{x}$ is input into a fully connected layer to obtain the node embedding vector $\boldsymbol{e}$.

$$\boldsymbol{e}_j = \boldsymbol{W}_j \boldsymbol{x}_j + \boldsymbol{b}_j, \tag{4}$$

where $\boldsymbol{W}_j \in \mathcal{R}^{d \times N}$ and $\boldsymbol{b}_j \in \mathcal{R}^d$ are learnable parameter matrices and biases, respectively, and d denotes the embedding dimension.

The diffusion dependence attention mechanism incorporates independent attention, which can make certain nodes dominate the attention weights. This dominance by a few influential nodes may neglect the information from other nodes, resulting in an incomplete reflection of the underlying diffusion network structure. We draw inspiration from coverage models used in linguistics. We utilize a coverage vector to keep track of previous attention results, enabling better guidance for future attention learning.

At each step, the process continues from the sampling start point to the endpoint along each sampled path. For a given path $\boldsymbol{M}_{p_{u\to v}}$, the nodes $N^{p_{u\to v}}$ from u to v are denoted as $\{n_0, n_1, \ldots, n_{l-1}\}$ in sequence, where l is the length of the path, and n_j and n_k are in the sequence. The structure of the diffusion-dependent attention layer with coverage mechanism is illustrated in Fig. 2. The calculation process is as follows:

$$\boldsymbol{e}_{k,j} = \boldsymbol{v}_e{}^T \tanh\left(\boldsymbol{W}_{e_1}\boldsymbol{e}_k + \boldsymbol{W}_{e_2}\boldsymbol{e}_j + \boldsymbol{W}_{e_3}\boldsymbol{V}_{k,j}\right), \tag{5}$$

$$a_{k,j} = \mathrm{softmax}\left(e_{k,j}\right), \tag{6}$$

$$\begin{aligned}\boldsymbol{V}_{k,j} &= \sigma\left(\boldsymbol{V}_{k-1,j}, \alpha_{k-1,j}, e_k, e_j\right) \\ &= \mathrm{sigmoid}(\boldsymbol{W}_{v_1}\boldsymbol{V}_{k-1,j} + \boldsymbol{W}_{v_2}\alpha_{k-1,j} \\ &\quad + \boldsymbol{W}_{v_3}\boldsymbol{e}_k + \boldsymbol{W}_{v_4}\boldsymbol{e}_j + \boldsymbol{b}_v),\end{aligned} \tag{7}$$

where $\boldsymbol{W}_{v_2} \in \mathcal{R}^d, \boldsymbol{W}_{e_1}, \boldsymbol{W}_{e_2}, \boldsymbol{W}_{e_3}, \boldsymbol{W}_{v_1}, \boldsymbol{W}_{v_3}$ and $\boldsymbol{W}_{v_4} \in \mathcal{R}^{d\times d}$ are parameter matrices, $\boldsymbol{v}_e, \boldsymbol{b}_v \in \mathcal{R}^d$ are parameter vectors. $\boldsymbol{V}_{k,j} \in \mathcal{R}^d$ denotes the newly introduced coverage vector, which summarizes the previous attention results. By combining the coverage vector from the previous steps and the diffusion dependence weights, the coverage vector enhances the memory properties of the attention mechanism, making the attention at each step no longer independent of each other. The diffusion-dependent node representation information is aggregated by weighted sum as follows:

$$\boldsymbol{d}_j = \sum_{k=1}^{j-1} \alpha_{k,j}\boldsymbol{e}_k. \tag{8}$$

The $\boldsymbol{d}_j$ is the node n_j's diffusion-dependent feature information. For each node n_j in the path sequence, the gate mechanism is applied to fuse its own embedding representation $\boldsymbol{e}_j$ with the diffusion-dependent feature information $\boldsymbol{d}_j$, resulting in node-level features with dependence-awareness, denoted as the non-sequential structural node representation $\boldsymbol{r}_j$:

$$\boldsymbol{r}_j = \boldsymbol{g}_j \odot \boldsymbol{e}_j + (1 - \boldsymbol{g}_j) \odot \boldsymbol{d}_j, \tag{9}$$

$$\boldsymbol{g}_j = \sigma\left(\boldsymbol{e}_j, \boldsymbol{d}_j\right) = \mathrm{sigmoid}\left(\boldsymbol{W}_{g_1}\boldsymbol{e}_j + \boldsymbol{W}_{g_2}\boldsymbol{d}_j + \boldsymbol{b}_g\right), \tag{10}$$

where $\boldsymbol{W}_{g_1}, \boldsymbol{W}_{g_2} \in \mathcal{R}^{d\times d}$, $\boldsymbol{b}_g \in \mathcal{R}^d$. The learned activation function $\boldsymbol{g}_j$ serves as a gate, allowing the discarding of unimportant information from its own embedding representation $\boldsymbol{e}_j$, while incorporating useful information from the diffusion-dependent feature information $\boldsymbol{d}_j$. This process enables the fused node representation $\boldsymbol{r}_j$ to be aware of its own diffusion dependence. In addition, we learn the node's sequential structural representation through LSTM.

$$\boldsymbol{s}_j = LSTM\left(\boldsymbol{e}_j, \boldsymbol{s}_{j-1}\right). \tag{11}$$

Subsequently, the gate mechanism is used once more to combine the node's non-sequential structural representation with the sequential structural representation, resulting in the final encoding of the node as follows:

$$\boldsymbol{h}_j = \boldsymbol{q}_j \odot \boldsymbol{s}_j + (1 - \boldsymbol{q}_j) \odot \boldsymbol{r}_j, \tag{12}$$

$$\boldsymbol{q}_j = \mathrm{sigmoid}\left(\boldsymbol{W}_{q_1}\boldsymbol{s}_j + \boldsymbol{W}_{q_2}\boldsymbol{r}_j + \boldsymbol{b}_q\right), \tag{13}$$

where $\boldsymbol{W}_{q_1}, \boldsymbol{W}_{q_2} \in \mathcal{R}^{d\times d}$, and $\boldsymbol{b}_q \in \mathcal{R}^d$. This allows the final fused node representation h_j to more comprehensively perceive and represent the features of sequential nodes. In the end, we calculate the average weighted sum of the nodes along the path to obtain the final path encoding.

$$\boldsymbol{M}_{p_{u\to v}} = \frac{1}{|N^{p_{u\to v}}|} \sum_{j \in N^{p_{u\to v}}} \boldsymbol{h}_j. \tag{14}$$

The GAIPSRec model with the coverage mechanism is denoted as GAIPSRec-cover.

3.3 Path Aggregation

For each path length, GAIPSRec samples several paths. In this part, we employ the attention mechanism inspired by the GAT framework to aggregate paths with the same length to construct subgraphs, allowing the model to learn the heterogeneous interaction information obtained by the information propagation path encoder. We generate attention weights by aligning the path representations of corresponding paths with the central node embedding.

$$\boldsymbol{e}_{uv}^{p} = \text{sigmoid}\left(\boldsymbol{W}_{\alpha^P}\left[\boldsymbol{h}_u \parallel \boldsymbol{M}_{p_{u\to v}}\right]\right), \tag{15}$$

$$\alpha_{uv}^{p} = \text{softmax}\left(\boldsymbol{e}_{uv}^{p}\right), \tag{16}$$

$$\boldsymbol{h}_u^{p} = \sum_{v\in N_u^p} \alpha_{uv}^{p} \boldsymbol{M}_{p_{u\to v}}, \tag{17}$$

where $\boldsymbol{e}_{uv}^{p}$ denotes the importance of the path $p_{u\to v}$ to the node u, and node u can be either a user or an item. $\parallel$ denotes concatenation. The attention weight α_{uv}^{p} denotes the importance of different paths to user nodes and the N_u^p denotes the set of nodes at distance p from node u. To reduce the potential high variance during the model training process, the multi-head attention mechanism [Vaswani *et al.*, 2017] is employed. We use k parallel attention modules to aggregate paths in a subgraph calculated by eq.17. Finally, the outputs from the k independent computations are combined.

$$\boldsymbol{h}_u^{p} = \frac{1}{K}\sum_{k=1}^{K}\sum_{v\in N_u^p}\left[\alpha_{uv}^{p}\right]_k \cdot \boldsymbol{M}_{p_{u\to v}}. \tag{18}$$

3.4 Subgraph Representation Aggregation

After constructing subgraphs, subgraph representation aggregation is performed. Although few researchers have explored subgraphs generated by the paths whose lengths are greater than 3, GAIPSRec can effectively learn information from higher-order neighbors, enriching the feature information of users and items. The attention mechanism in GAIPSRec adaptively learns the importance of various subgraphs and aggregates the embedding representations of different subgraphs through weighted sum to update the central node.

For all nodes of the same type (users or items), their subgraph representations with the same path length ($\boldsymbol{h}_u^{p}$) are subjected to linear and nonlinear transformation, and then the average is taken to obtain the ensemble representation of the subgraphs for that specific length. For instance, for users, the process is as follows:

$$\boldsymbol{h}_U^{P_i} = \frac{1}{|U|}\sum_{u\in U}\sigma\left(\boldsymbol{W}_{P_i}\boldsymbol{h}_u^{p_i}\right), \tag{19}$$

where P_i denotes the set of paths with a specific length, and each path instance $p_i \in P_i$. U denotes the set of users, and u denotes a specific user node. $\mathbf{P}$ denotes the set of path lengths for users or items, and $P_i \in \mathbf{P}$. The alignment of representations for paths of different lengths is achieved through linear transformation, and then softmax normalization is applied to obtain attention weights. For user nodes, the calculation process is as follows, and item nodes are similar.

$$\boldsymbol{e}_U^{P_i} = \boldsymbol{W}_{\alpha_U^{P_i}}\boldsymbol{M}_U^{P_i}, \tag{20}$$

$$\alpha_U^{P_i} = \text{softmax}\left(\boldsymbol{e}_U^{P_i}\right), \tag{21}$$

$$\boldsymbol{h}_u = \sum_{P_i\in P}\alpha_U^{P_i}\boldsymbol{h}_u^{P_i}. \tag{22}$$

3.5 Rating Prediction

Inspired by the idea of residual connections, GAIPSRec incorporates connections by concatenating the node's original embedding with its final embedding to obtain the model's output, which can alleviate the over-smoothing problem.

$$\boldsymbol{h}'_u = \boldsymbol{W}_{final_u}\left(\boldsymbol{h}_u \parallel \boldsymbol{e}_u\right). \tag{23}$$

Then we concatenate user embedding representations with item embedding representations, employing a multi-layer linear network to predict the ratings.

$$\hat{r}_{ui} = MLP\left(\boldsymbol{h}'_u \parallel \boldsymbol{h}'_i\right). \tag{24}$$

3.6 Model Training

In our paper, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are combined and used as the loss function for model training. Additionally, L_2 regularization is applied to the model parameters to prevent over-fitting.

$$L_r = \sum_{(u,i)}\left[\left(\hat{r}_{ui} - r_{ui}\right)^2 + \left|\hat{r}_{ui} - r_{ui}\right|\right] + \lambda\sum_{i=1}^{n}\Theta, \tag{25}$$

where the λ denotes the penalty coefficient for regularization, and Θ corresponds to the model parameters. The Adam optimizer is used to train GAIPSRec.

Rank learning aims to present a sorted array of items to users, a common approach in various recommendation systems like top-k and sequential recommendations. For every user, items they have interacted with are marked as 1 (positive samples) when their rating meets or exceeds a threshold F, or 0 (negative samples) otherwise. This implies that users are inclined to engage with items rated F or higher. Considering our datasets have ratings from 1 to 5, we conduct experiments with $F = 3$. Model predictions are refined using the sigmoid function, and for the task of ranking, the loss is computed using the Binary Cross-Entropy (BCE) Loss:

$$L_b = -\sum_{(u,i)} y_{ui}\log\left(\hat{y}_{ui}\right) + \left(1 - y_{ui}\right)\log\left(1 - \hat{y}_{ui}\right), \tag{26}$$

4 Experimental Results

4.1 Experimental Settings

Datasets. The real-world recommendation datasets Ciao[Fan *et al.*, 2019b] and Epinions[Fan *et al.*, 2019a] are selected for the experiments. For each dataset, 80% of the rating data are randomly sampled to form the training set, while the remaining 20% are used as the test set. Table 1 presents the relevant information for these datasets.

Evaluation Metrics. To evaluate the model's rating performance, we use the MAE and RMSE. Smaller values of MAE and RMSE indicate better performance. We use the Precision and NDCG to evaluate the ranking performance, and the bigger values indicate better performance.

	Ciao	Epinions
Number of users	7,375	22,166
Number of items	106,797	296,277
Number of ratings	284,086	922,267
Rating density	0.0361%	0.0140%
Number of relationships of trust	111,781	355,813
Relational density of trust	0.2055%	0.0724%

Table 1: Ciao and Epinions dataset information.

Baselines. This paper selects the following representative models to compare their performance with GAIPSRec:

- Traditional social recommendation algorithm: **TrustMF** [Yang *et al.*, 2016], **TrustSVD** [Guo *et al.*, 2016].
- Deep neural network based recommendation algorithms: **SSL-SVD** [Hu *et al.*, 2020], **NeuCDCF** [Vijaikumar *et al.*, 2021], **ConsisRec** [Yang *et al.*, 2021].
- Graph neural network based recommendation algorithms: **GraphRec** [Fan *et al.*, 2019a], **SCSVD** [Guan *et al.*, 2021], **GDSRec** [Chen *et al.*, 2023], **MI3Graph** [Zhu *et al.*, 2023].

Parameter Settings. In order to compare different methods, we consider the parameter configurations by cross-validation. To ensure fairness in our comparison, all experimental results are reported as the average of five independent runs. In this paper, we use PyTorch and Dgl to implement the GAIPSRec model. The learning rate η is set to 10^{-4}, and the embedding dimension D is 64. The hidden layer dimension is the same as the input layer dimension, and all parameters are initialized using the Xavier uniform distribution. For both user and item nodes, we sample paths of lengths [2, 3, 4, 5], and the dropout rate is set to 0.5. We test the restart probability p_{rwr} is set to 0.2.

4.2 Comparative Experiments

In this part, we conduct comprehensive experiments to compare the proposed GAIPSRec model with existing models. The experimental results of GAIPSRec and the baseline models are presented in Table 2.

GraphRec performs better than TrustMF and TrustSVD, probably because of the utility of performing propagation information across users and items under a graph-structured neural network. GDSRec's good performance on the Ciao dataset is attributed to the fact that it utilizes both user and program statistics. MI3Graph beats most social recommendation models on the Epinions dataset due to its incorporation of multi-interaction information. GAIPSRec-cover achieves the best results on both the Ciao dataset and the Epinions dataset. Compared to the best-performing baseline model, GAIPSRec shows improvements of 3.28% and 1.64% in MAE and RMSE, respectively, on the Ciao dataset. On the Epinions dataset, GAIPSRec achieves improvements of 2.31% and 0.88% in MAE and RMSE, respectively. These results demonstrate that the model design of GAIPSRec, which combines random walk simulations for information propagation with path aggregation, is effective. Moreover, GAIPSRec-lstm outperforms GAIPSRec-gru, but both models show similar performance. The LSTM encoder, with its more complex network structure, has stronger representation capabilities. However, the mean encoder, GAIPSRec-mean, exhibits weaker representation capabilities compared to RNN-based encoders.

	Ciao		Epinions	
	MAE (%)	**RMSE (%)**	**MAE (%)**	**RMSE (%)**
TrustMF	76.90	104.79	84.10	113.95
TrustSVD	77.01	103.56	83.87	107.31
GraphRec	73.87	97.94	81.68	106.31
SSL-SVD	73.31	98.90	80.16	104.55
NeuCDCF	73.24	98.35	80.52	105.09
ConsisRec	73.94	97.40	80.46	104.95
SCSVD	73.64	97.61	79.98	104.38
GDSRec	<u>73.23</u>	<u>97.22</u>	80.47	105.66
MI3Graph	73.53	97.81	<u>79.95</u>	<u>104.29</u>
GAIPSRec-cover	**70.83**	**95.63**	**78.10**	**103.37**
GAIPSRec-lstm	71.25	96.18	78.68	103.87
GAIPSRec-gru	74.01	97.17	80.81	103.66
GAIPSRec-mean	74.38	97.26	80.90	103.76

Table 2: MAE and RMSE of GAIPSRec and baseline models.

4.3 Item Ranking

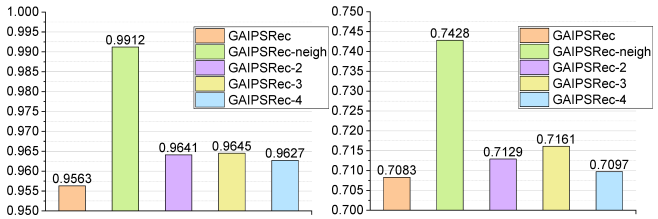
In this part, we evaluate the item ranking performance of our model on Ciao and Epinions. We selected several top-n ranking algorithms for comparison: **PMF** [Mnih and Salakhutdinov, 2007], **SVD++**, **NeuMF**, **LightGCN**, **GraphRec**, **Diffnet++**, and **GDSRec**. The results are depicted in Table 3. It is observed that the performance differences on the Precision metric are minimal. This is attributed to the high proportion of positive samples in the dataset, resulting in generally high Precision across all models. By contrast, there is a noticeable variance in model performance on the NDCG metric, where GAIPSRec outperforms the other models. This indicates that GAIPSRec is more effective in promoting positive items to higher ranks.

4.4 Ablation Study

In this part, we conduct an ablation study to investigate the effectiveness of different modules in GAIPSRec. By comparing GAIPSRec with its variants, namely GAIPSRec-2,

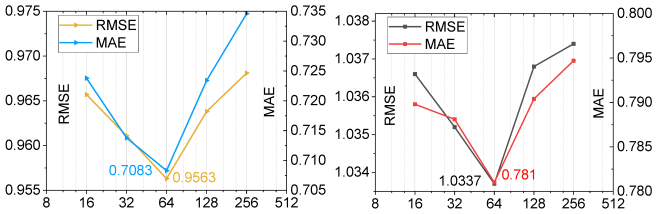
	Ciao		Epinions	
	Precision (%)	**NDCG (%)**	**Precision (%)**	**NDCG (%)**
PMF	98.71	93.80	98.01	90.30
SVD++	98.71	94.22	97.97	91.00
NeuMF	98.70	94.10	98.01	90.91
GCMC	98.69	94.51	98.00	92.12
LightGCN	98.66	95.30	<u>98.01</u>	93.71
GraphRec	<u>98.72</u>	95.73	97.99	93.69
Diffnet++	98.67	95.85	97.96	93.62
GDSRec	98.69	<u>96.10</u>	97.98	<u>94.20</u>
GAIPSRec	**98.77**	**96.60**	**98.05**	**95.70**

Table 3: Performance of ranking on Ciao and Epinions.



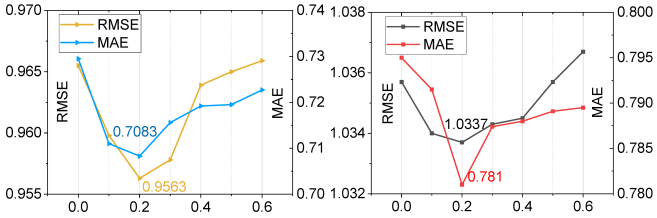
(a) RMSE (b) MAE

Figure 3: GAIPSRec and its variants.



(a) Ciao (b) Epinions

Figure 4: Effect of dimensionality D.



(a) Ciao (b) Epinions

Figure 5: Effect of restart probability p_{rwr}.

GAIPSRec-3, GAIPSRec-4, and GAIPSRec-neigh, we can assess the importance of each module. GAIPSRec-L removes the paths of length $\boldsymbol{L}$ and their corresponding subgraphs. GAIPSRec-neigh does not use random walk-sampled paths. Instead, it only aggregates path neighbors and performs subgraph representation aggregation on the subgraphs to demonstrate the effectiveness of information propagation modeling.

Here, we only present the experimental results on the Ciao dataset, and the results on the Epinions dataset are similar. As shown in Fig. 3, GAIPSRec-neigh does not utilize path information propagation encoding and achieves the worst results on both of the datasets. Other variants that combine information propagation modeling with path aggregation outperform GAIPSRec-neigh, indicating that the proposed random walk path sampling and path information encoding significantly improve the performance, and the combination of information propagation and path aggregation is effective.

Regarding the variants that remove paths of certain lengths and their corresponding subgraphs, on both datasets, GAIPSRec-4 performs the best, indicating that removing the subgraphs of 4th-order neighbors has relatively little impact on the model. Additionally, GAIPSRec-2 slightly outperforms GAIPSRec-3. One possible explanation is that 3rd-order neighbors are more likely to be of the same type, such as item-item relationships, which have a larger impact on the model. All variants perform worse than the complete GAIPSRec, validating the effectiveness of each part in the model.

4.5 Hyperparameter Analysis

In this part, we conduct experiments to analyze the impact of hyperparameters on the model, including the embedding dimension D, and the random walk restart probability p_{rwr}. The experiments in this section are performed using the GAIPSRec-cover model.

We conduct experiments using embedding dimensions [16, 32, 64, 128, 256]. As illustrated in Fig. 4, on both datasets, the optimal RMSE and MAE are obtained with an embedding dimension of 64. When the dimension is lower, the model's expressive power still has potential for improvement. Conversely, with higher dimensions, over-fitting may occur, leading to a drop in performance.

GAIPSRec leverages random walk with restart for path sampling to approximate the information propagation process. Restarting the random walk enhances the influence of the start node. In this part, we experiment with $p_{rwr} \in [0, 0.6]$. As depicted in Fig. 5, the model on both datasets attains optimal performance with the restart probability of 0.2.

When the restart probability is small, an increase in p_{rwr} facilitates the sampling of paths that are nearer to the center node, resulting in a decrease in both RMSE and MAE. Conversely, when the restart probability is high, an increase in p_{rwr} causes that fewer neighboring nodes are included.

5 Conclusions

This paper presents an graph attention network with information propagation and path aggregation for social recommendation. We perform sampling based on proposed universal framework for heterogeneous graph sampling, eliminating the need for manually defining meta-paths. This sampling method helps to facilitate the acquisition of information from higher-order neighbors, and alleviate the prevalent problem of over-smoothing within the GNNs and the constraints brought about by the definition of meta-paths. Our information propagation path encoding consider all the nodes along the path. Additionally, the coverage mechanism encoder with sequential dependence is used to model the information propagation process and gains a more holistic view of the data and reflects the potential diffusion network structure. We combines information propagation with path aggregation in GNNs. Subgraph representation aggregation is performed in subgraphs composed of path instances, providing users and items with richer high-order neighbor and graph structure information. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed model.

Acknowledgments

This research is supported by the Natural Science Foundation of Shaanxi Province for Distinguished Young Scholars (No. 2023-JC-JQ-54).

References

[Bruna *et al.*, 2013] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[Chen *et al.*, 2018] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[Chen *et al.*, 2023] Jiajia Chen, Xin Xin, Xianfeng Liang, Xiangnan He, and Jun Liu. Gdsrec: Graph-based decentralized collaborative filtering for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4813–4824, 2023.

[Chiang *et al.*, 2019] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 257–266, 2019.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[Fan *et al.*, 2018] Wenqi Fan, Qing Li, and Min Cheng. Deep modeling of social relations for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Fan *et al.*, 2019a] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[Fan *et al.*, 2019b] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. Deep social collaborative filtering. In *Proceedings of the 13th ACM conference on recommender systems*, pages 305–313, 2019.

[Fatemi and Tokarchuk, 2013] Maryam Fatemi and Laurissa Tokarchuk. A community based social recommender system for individuals & groups. In *2013 International Conference on Social Computing*, pages 351–356. IEEE, 2013.

[Guan *et al.*, 2021] Jiewen Guan, Xin Huang, and Bilian Chen. Community-aware social recommendation: A unified scsvd framework. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Guo *et al.*, 2016] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel recommendation model regularized with user trust and item ratings. *ieee transactions on knowledge and data engineering*, 28(7):1607–1620, 2016.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2354–2366, 2018.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[Hu *et al.*, 2020] Zhengdi Hu, Guangquan Xu, Xi Zheng, Jiang Liu, Zhangbing Li, Quan Z Sheng, Wenjuan Lian, and Hequn Xian. Ssl-svd: semi-supervised learning–based sparse trust recommendation. *ACM Transactions on Internet Technology (TOIT)*, 20(1):1–20, 2020.

[Huang *et al.*, 2021] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. Knowledge-aware coupled graph neural network for social recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4115–4122, 2021.

[Jamali and Ester, 2010] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142, 2010.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[Lin *et al.*, 2022] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 2320–2329, 2022.

[Liu *et al.*, 2018] Chun-Yi Liu, Chuan Zhou, Jia Wu, Yue Hu, and Li Guo. Social recommendation with an essential preference space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Liu *et al.*, 2020] Fanzhen Liu, Shan Xue, Jia Wu, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Jian Yang, and Philip S Yu. Deep learning for community detection: progress, challenges and opportunities. *arXiv preprint arXiv:2005.08225*, 2020.

[Ma *et al.*, 2008] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940, 2008.

[Mnih and Salakhutdinov, 2007] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

[Shi *et al.*, 2018] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[Tang *et al.*, 2013] Jiliang Tang, Xia Hu, and Huan Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3:1113–1133, 2013.

[Tang *et al.*, 2016] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang, and Huan Liu. Recommendation with social dimensions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Vijaikumar *et al.*, 2021] M Vijaikumar, Shirish Shevade, and M Narasimha Murty. Neural cross-domain collaborative filtering with shared entities. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, pages 729–745. Springer, 2021.

[Wang *et al.*, 2019a] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[Wang *et al.*, 2019b] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.

[Wu *et al.*, 2019a] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng 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*, pages 235–244, 2019.

[Wu *et al.*, 2019b] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The world wide web conference*, pages 2091–2102, 2019.

[Wu *et al.*, 2020] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4753–4766, 2020.

[Xing *et al.*, 2022] Su Xing, Xue Shan, Liu Fanzhen, Wu Jia, Yang Jian, Zhou Chuan, Hu Wenbin, Paris Cecile, Nepal Surya, Jin Di, et al. A comprehensive survey on community detection with deep learning. *IEEE Trans. Neural Netw. Learn. Syst*, 2022.

[Yang *et al.*, 2016] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1633–1647, 2016.

[Yang *et al.*, 2021] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S Yu. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*, pages 2141–2145, 2021.

[Zhu *et al.*, 2023] Haorui Zhu, Fei Xiong, Hongshu Chen, Xi Xiong, and Liang Wang. Incorporating a triple graph neural network with multiple implicit feedback for social recommendation. *ACM Transactions on the Web*, 2023.