Social Influence Learning for Recommendation Systems

Ximing Chen
University of Macau
Macau SAR, China
yc37921@um.edu.com

Pui Ieng Lei
University of Macau
Macau SAR, China
yc37460@um.edu.mo

Yijun Sheng
University of Macau
Macau SAR, China
yc17419@um.edu.mo

Yanyan Liu
University of Macau
Macau SAR, China
yc07402@um.edu.mo

Zhiguo Gong*
University of Macau
Macau SAR, China
fstzgg@um.edu.mo

Abstract

Social recommendation systems leverage the social relations among users to deal with the inherent cold-start problem in user-item interactions. However, previous models only treat the social graph as the static auxiliary to the user-item interaction graph, rather than dig out the hidden essentials and optimize them for better recommendations. Thus, the potential of social influence is still under-explored. In this paper, we will fill this gap by proposing a novel model for social influence learning to derive the essential influence patterns within the user relationships. Our model views the social influence from the perspectives of (1) the diversity of neighborhood's influence on the users, (2) the disentanglement of neighborhood's influence on the users, and (3) the exploration of underlying implicit social influence. To this end, we first employ a novel layerwise graph-enhanced variational autoencoder for the reconstruction of neighborhoods' representations, which aims to learn the pattern of social influence as well as simulate the social profile of each user for overcoming the sparsity issue in social relation data. Meanwhile, we introduce a layerwise graph attentive network for capturing the most influential scope of neighborhood. Finally, we adopt a dual sampling process to generate new social relations for enhancing the social recommendation. Extensive experiments have been conducted on three widely-used benchmark datasets, verifying the superiority of our proposed model compared with the representative approaches.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Social Recommendation, Social Influence, generative models, graph convolution networks

ACM Reference Format:
Ximing Chen, Pui Ieng Lei, Yijun Sheng, Yanyan Liu, and Zhiguo Gong. 2024. Social Influence Learning for Recommendation Systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679598

1 Introduction

Social recommendation is one type of recommendation systems that leverages the social network among users to extract personalized information on individuals. According to the study of homophily in social networks [26, 39], recommendation systems can be modeled not only based on users' purchase records but also their social relations. As commonly known, people may make decisions under the influence of their relatives. This is the so-called *social influence*. Regarding the types of social influence, it is worth noting that *explicit social influence* should be direct, trustworthy, and acknowledged connections among users, while *implicit social influence* is the potential connections between users having similar preferences. In most scenarios, *explicit social influence* can be retrieved from the relational data on social media. Previous works [8, 14, 23] have shown the outperformance of social recommendation systems as compared to general recommendation systems.

While social recommendation has the capability to deal with personalization and correlations, there are still many challenges to address. Existing models mostly take the social relation data as the static auxiliary to the user-item interaction data, and they simply combine the data to handle the cold-start problem in recommendation. However, the shallow links given by the social relations may not reveal the essential influences among users. This problem is totally ignored by those previous works. Overall, the following three challenges need to be attached importance:

Explicit Influence I - Social Influence Diversity: Users may connect with each other through various relationships (e.g. friends, family, and fellows, as shown in Figure 1(a)). Different types of relationships should have different patterns of influence passing. However, owing to the privacy issues, researchers face a lack of labels on relationships. Therefore, designing a fine-drawn model to disentangle the relationships is a de facto challenge. It is commonly believed that users of the same relationships should exhibit similar behaviors. In that case, we can simply cluster socially connected users that have bought the same items, and assign them with the same pseudo-relational labels. Nevertheless, users' social behaviors are much more elusive than we thought. The inconsistency and incompleteness of the original social network, conceived as social influence diversity, ought to be resolved on an ad hoc basis.

*Corresponding author.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679598

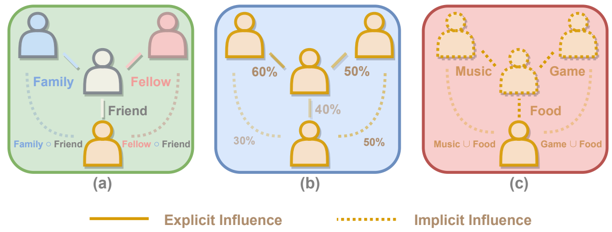

Figure 1: Toy example of (a) Social Influence Diversity, (b) Social Influence Propagation, and (c) Social Influence Exploration. For (b), the values denote the pseudo-consistency between users with each other.

Explicit Influence II - Social Influence Propagation: Within the social network, users are interconnected as a whole. Under the scheme of GNN-based models, users can aggregate from a larger fraction of users that are not even directly connected. The challenge of social influence propagation is that different magnitudes of social influence exist on the graph. Moreover, users in closer neighborhoods may not always have higher social influence (as shown in Figure 1(b)). Apart from local social influence that usually occurs in nearby neighborhoods, global social influence from larger neighborhoods is also nonnegligible. Hence, instead of the proximity on graph, the magnitude of social influence propagation should depend on the consistency of behavior between the user and the neighborhoods, with higher consistency implying stronger social influence, and vice versa.

Implicit Influence - Social Influence Exploration: Social influence is introduced to enhance the recommendation since it mitigates the data sparsity problem of user-item interactions. However, unlike user-item interactions, social influence is weak-tie relations, making it treacherous to be used to predict the strong-tie relations on the interaction graph. This impels us to work the other way around — that is, to discover connections within the social network using the interaction graph. Through user-item interactions, users are thought to be making implicit influences on those who have bought similar items (as Figure 1(c)). This makes them potential neighbors on the social network. However, simply connecting these users on the social network is irrational and could cause user representation collapse. Moreover, it is a chicken and egg problem to determine the social network together with user representations.

To address these challenges, we propose **E**xplicit and **I**mplicit **I**nfluence **S**ocial **R**ecommendation **S**ystem (EIISRS). EIISRS consists of three corresponding stages: layerwise graph-enhanced variational autoencoder, layerwise graph attention networks, and dual sampling process. Our model is a two-tower training architecture, which includes a social path and a bipartite path. The social network and the user-item interaction graph are separately feed-forward to train two different types of user representations. This allows the model to automatically balance between two paths and avoid data entanglement. To perform social influence learning, we first apply Graph Convolutional Networks (GCN) to generate social user representations. Unlike the previous works, we take the social user representations generated from different layers of convolution for the training, so that we can model different scope of neighborhood separately and disentangle social influence. Next, the representation of each layer is fed into different Variational Autoencoders (VAEs) to learn the distribution of neighborhoods, which contains the social patterns. Meanwhile, to simulate the social influence diversity, the reparameterization trick is used to sample neighborhood representations. The representations of different layers are then fused by a layerwise graph attention network for modeling the different magnitudes of social influence. Finally, to generate augmented neighborhoods for social influence exploration, a dual sampling (Gumbel sampling followed by Bernoulli sampling) is applied to prevent user representation collapse. Overall, the major contributions of this paper are summarized as follows:

- We analyze and sort out three different challenges in social recommendation systems, and propose three novel components to deal with them accordingly.
- We propose a GCN-based two-tower architecture to incorporate a layerwise graph-enhanced variational autoencoder, a layerwise graph attention network, and a dual sampling process. To the best of our knowledge, we are the first to combine these methods for social network disentanglement in social recommendation systems.
- We conduct extensive experiments on multiple real-world datasets to demonstrate the superiority of the proposed model and the effectiveness of each component.

2 Methodology

In this section, we introduce our model **EIISRS** in detail. We will elaborate on our framework (as Figure 2) and three components used to solve the respective challenges of social influence modeling.

2.1 Preliminaries

Let u denote the user and i denote the item within the dataset. $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ represents the entire user set and $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ represents the entire item set, where the total numbers of users and items are $|\mathcal{U}| = m$ and $|\mathcal{I}| = n$, respectively. $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ is the binary user-item interaction matrix. Given a user-item pair (u, i), $r_{u,i} = 1$ implies that user u purchased item i and should be interested in it, whereas $r_{u,i} = 0$ implies no interaction. $\boldsymbol{S} \in \mathbb{R}^{m\times m}$ refers to the social matrix, whose entries are 1 if the corresponding users are socially connected and 0 otherwise. We build a bipartite graph $\mathcal{G}_r = \{(u, r_{u,i}, i)|u \in \mathcal{U}, i \in \mathcal{I}, r_{u,i} \in \{0, 1\}\}$ from the user-item interaction matrix $\boldsymbol{R}$. Likewise, we build a social graph $\mathcal{G}_s$ from the social matrix $\boldsymbol{S}$. $\mathcal{I}(u)$ is the item set containing the items purchased by user u. $\mathcal{N}(u)$ is the set of neighboring users who are directly connected to user u on the social network. Each convolution layer in our model generates neighborhood representations. These representations are denoted as $\{\boldsymbol{P}^{(0)}, \boldsymbol{P}^{(1)}, \ldots, \boldsymbol{P}^{(L)}\}$ for users and $\{\boldsymbol{Q}^{(0)}, \boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(L)}\}$ for items, where $\boldsymbol{P}^{(l)} \in \mathbb{R}^{m\times d}$ and $\boldsymbol{Q}^{(l)} \in \mathbb{R}^{n\times d}$ represent the l-th layer embeddings of d-dimensional size. In our model, we focus on top-K recommendations, and $\hat{r}_{u,i}$ denotes the computed likelihood that user u is interested in item i, i.e., the recommendation score. We use bold capital letters to denote matrices and bold lowercase letters to denote vectors. Note that $\mathcal{N}(\cdot)$ refers to the set of neighbors, while $\boldsymbol{\mathcal{N}}(\cdot)$ in boldface refers to the normal distribution.

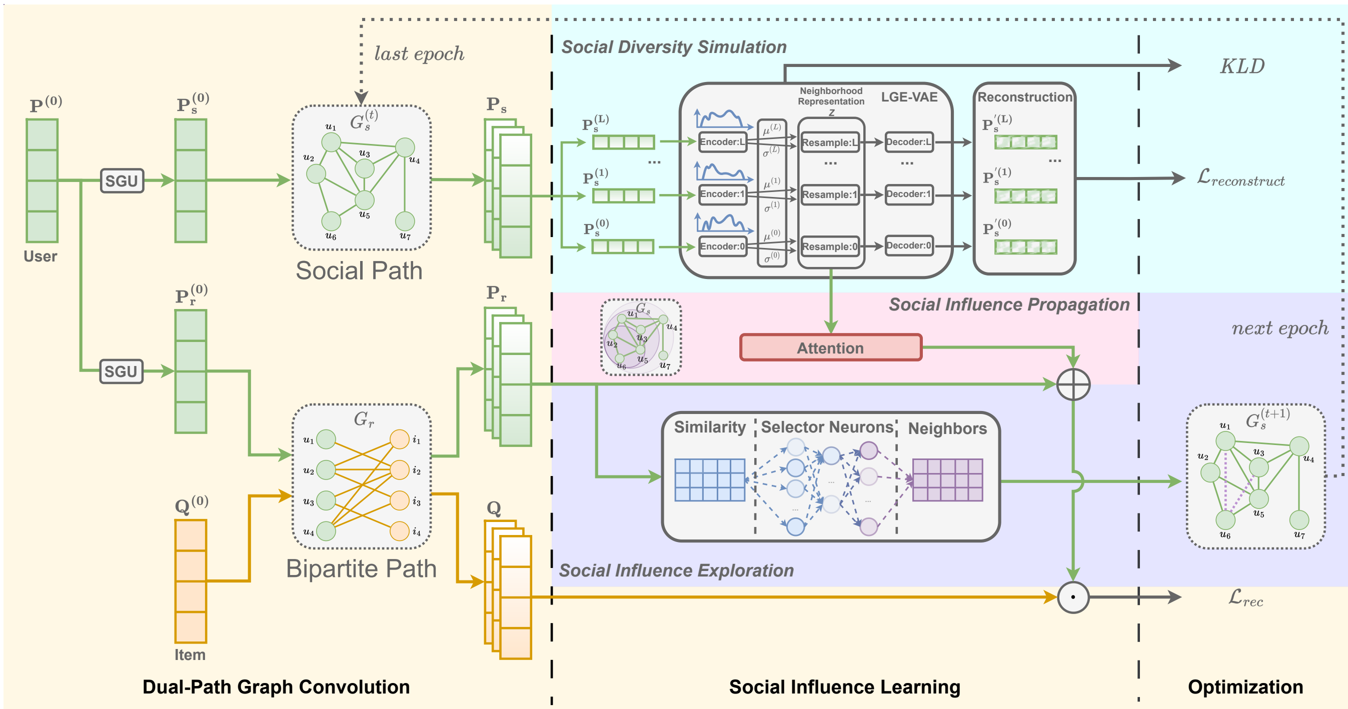

Figure 2: The framework of EIISRS.

2.2 Dual-Path Graph Convolution

In our model, we use a two-tower setting, including the bipartite path and the social path. The bipartite path takes $\mathcal{G}_r$ as input and jointly encodes user and item representations. The social path takes $\mathcal{G}_s$ as input and encodes another set of user representations. Since the patterns on different graphs are of different importance to the representation learning, we need to control how much information to feed forward in each path. To achieve that, we propose to preprocess the base user representations $\boldsymbol{P}^{(0)}$ with dual partial maskings, which can be learned using the filtering Self-Gating Units (SGUs) [3] that adopt the idea called multiplicative skip connection:

$$\boldsymbol{P}^{(0)}_{path} = f_{sgu}(\boldsymbol{P}^{(0)}) = \boldsymbol{P}^{(0)} \odot \sigma(\boldsymbol{P}^{(0)} \boldsymbol{W}^{path}_{g} + \boldsymbol{b}^{path}_{g}), \quad (1)$$

where $\boldsymbol{W}^{path}_{g} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}^{path}_{g} \in \mathbb{R}^{d}$ are learnable parameters, $path \in \{r, s\}$ denotes the path, $\odot$ represents the Hadamard product, and σ is the nonlinear activation function. Here we use the sigmoid function as the activation since it squashes the values into $[0, 1]$ for the gating.

Referring to the graph convolutional scheme proposed in [17], graph convolutions in our model can be rewritten as:

$$\boldsymbol{P}^{(l+1)}_{r} = \boldsymbol{D}^{-1}_{user} \boldsymbol{R} \boldsymbol{Q}^{(l)}, \quad (2)$$

$$\boldsymbol{Q}^{(l+1)} = \boldsymbol{D}^{-1}_{item} \boldsymbol{R}^{\top} \boldsymbol{P}^{(l)}_{r}, \quad (3)$$

$$\boldsymbol{P}^{(l+1)}_{s} = \boldsymbol{D}^{-1}_{s} \boldsymbol{S} \boldsymbol{P}^{(l)}_{s}, \quad (4)$$

where $\boldsymbol{P}^{(l)}_{r}$ and $\boldsymbol{P}^{(l)}_{s}$ are the l-th layer user representations generated from the user-item interaction graph and social graph, respectively. $\boldsymbol{D}_{user}$, $\boldsymbol{D}_{item}$, and $\boldsymbol{D}_{s}$ are the diagonal degree matrices of $\boldsymbol{R}$, $\boldsymbol{R}^{\top}$, and $\boldsymbol{S}$. We follow the guidance of LightGCN [11] and employ neighborhood information aggregations without feature transformations and nonlinear activation functions.

2.3 Social Influence Learning

After the elementary graph convolutions, the social influence learning phase follows to capture comprehensive social patterns for improved recommendations.

2.3.1 Social Influence Diversity. Social influence is diverse as users are interconnected over various relationships. Moreover, social networks are sparse. Only part of the links are observable while the rest remain unknown. Therefore, one way to optimize social network modeling is to apply link predictions and abnormal detections. However, it will be expensive to train such a model, and the model will need to comply with a pre-training/fine-tuning framework. We wish to use a lightweight and end-to-end scheme for capturing the social influence diversity.

To this end, we use VAE [15] as a building block and propose Layerwise Graph-Enhanced Variational AutoEncoder (LGE-VAE). Specifically, the input features of LGE-VAE are $\{\boldsymbol{P}^{(0)}_{s}, \boldsymbol{P}^{(1)}_{s}, \ldots, \boldsymbol{P}^{(L)}_{s}\}$, the layerwise neighborhood representations learned from previous social graph convolutions. LGE-VAE encodes these features as layerwise distributions over the latent space. Then, LGE-VAE samples a set of representations $\{\boldsymbol{Z}^{(0)}_{s}, \boldsymbol{Z}^{(1)}_{s}, \ldots, \boldsymbol{Z}^{(L)}_{s}\}$ for each user from the layerwise distributions. The uncertainty within the sampling process helps to capture the diversity in social influence. Finally, a decoder is used to reconstruct the input features.

Overall, the encoder can be viewed as the posterior distribution $p_\phi(\boldsymbol{Z}_s^{(l)}|\boldsymbol{P}_s^{(l)})$ with learnable parameters ϕ. Following the conventions, the intractable posterior distribution is approximated with a Gaussian distribution as the prior distribution, defined as follows:

$$q_\phi(\boldsymbol{Z}_s^{(l)}|\boldsymbol{P}_s^{(l)}) = \mathcal{N}(\mu(\boldsymbol{P}_s^{(l)}), \sigma^2(\boldsymbol{P}_s^{(l)})), \tag{5}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are the functions that compute the mean and standard deviation for the layerwise representations. Here, we assume that the Gaussian distribution is anisotropic. That is, the deviations of layerwise neighborhood distributions are different across users and dimensions. This helps to capture each user's unique interests and affection to items [45].

Specifically, the encoder of LGE-VAE generates the layerwise mean $\boldsymbol{\mu}^{(l)} \in \mathbb{R}^{m\times d}$ and the layerwise standard deviation $\boldsymbol{\sigma}^{(l)} \in \mathbb{R}^{m\times d}$. Given the distributions, we apply the reparameterization trick to sample new social neighborhood representations as below:

$$\boldsymbol{\mu}^{(l)}, \boldsymbol{\sigma}^{(l)} = MLP(\boldsymbol{P}_s^{(l)}, \boldsymbol{W}_{enc}), \tag{6}$$

$$\boldsymbol{Z}_s^{(l)} = \boldsymbol{\mu}^{(l)} + \boldsymbol{\sigma}^{(l)} \odot \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{7}$$

where $\boldsymbol{W}_{enc} \in \mathbb{R}^{d\times 2d}$ is the learnable parameter for the encoder, $MLP(\cdot) : \mathbb{R}^{m\times d} \times \mathbb{R}^{d\times 2d} \mapsto \mathbb{R}^{m\times 2d}$ is the Multilayer Perceptron (MLP), and $\boldsymbol{\epsilon} \in \mathbb{R}^{m\times d}$ is the Gaussian noise. This sampling process can be regarded as a simulation of social influence diversity. Then, we take the negation of the Evidence Lower Bound (ELBO) as the loss function of LGE-VAE, which is defined as follows:

$$\begin{aligned}\mathcal{L}_{vae}(\psi, \phi, \beta) = &-\mathbb{E}_{q_\phi}[\log p_\psi(\boldsymbol{P}_s^{(l)}|\boldsymbol{Z}_s^{(l)})] \\ &+ \beta KL(q_\phi(\boldsymbol{Z}_s^{(l)}|\boldsymbol{P}_s^{(l)})||p(\boldsymbol{Z}_s^{(l)})),\end{aligned} \tag{8}$$

where ϕ and ψ are learnable parameters of the encoder and decoder, respectively. The first term is the reconstruction loss while the second term is the KL divergence loss. Here we adopt β-VAE, where the hyperparameter β helps to discover disentangled latent factors. When $\beta \to 1$, β-VAE degrades to regular VAE, in which the KL divergence will take as much importance as the reconstruction loss. We set $\beta \ll 1$ to prioritize the reconstruction loss and stabilize the training of the encoder. It is worth noting that, LGE-VAE is different from VAE and VGAE in that it inputs neither the real-world attributes as features nor the entire graph structure.

2.3.2 Social Influence Propagation. The previous graph convolutions and social diversity simulations together generate a set of representations $\{\boldsymbol{Z}_s^{(0)}, \boldsymbol{Z}_s^{(1)}, \ldots, \boldsymbol{Z}_s^{(L)}\}$ for each user, where $\boldsymbol{Z}_s^{(l)}$ contains the social influence within the l-hop neighborhood. The question that follows immediately is how to aggregate these multilayer social representations. As shown in Table 1, there are four main categories of aggregation methods. For instance, mean pooling is the most commonly used one, which has been proven to be effective in preventing over-smoothing. However, it cannot well disentangle the social influence of distinct layers. Meanwhile, social influence should be propagated from a neighborhood of appropriate size, neither too large that consists of noise nor too small that results in collapse. To this end, we apply the attention mechanism [31] to aggregate the layerwise social representations. For each user u, the tuple $(\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L)$ contains the attention scores for each layer of representations. The function to compute the attention scores is f_{att}, defined as:

$$\alpha_l = f_{att}(\boldsymbol{Z}_s^{(l)}) = \frac{\exp(\boldsymbol{a}^\top \boldsymbol{W}_{att}\boldsymbol{Z}_s^{(l)})}{\sum_{j=0}^{L}\exp(\boldsymbol{a}^\top \boldsymbol{W}_{att}\boldsymbol{Z}_s^{(j)})}, \tag{9}$$

where both $\boldsymbol{a} \in \mathbb{R}^d$ and $\boldsymbol{W}_{att} \in \mathbb{R}^{d\times d}$ are trainable parameters. The final representation of user in social path after the aggregation is $\widetilde{\boldsymbol{P}}_s = \sum_{l=0}^{L}\alpha_l\boldsymbol{Z}_s^{(l)}$. In fact, explicit social influence is usually very noisy. Even shallow (e.g., 1-hop) propagations can contain much noise, making it hard to extract useful information therefrom [42, 43]. Recently, layerwise weighted graph network has shown effectiveness on graph representation learning [13, 21], but it has yet to find its place in recommendations. The LightGCN experiment [11] demonstrated the futility of layerwise graph attention for recommendations, as the user-item bipartite graph does not reflect informative patterns between layers. Different from previous works, we are the first to apply layerwise attention mechanism for social recommendation tasks. It helps to balance layerwise messages and filter out noise on the social graph, such that useful information can be extracted despite the noisiness in explicit social influence.

Table 1: Methods for aggregating layerwise representations. "Unified" denotes the integrated learning of social influence in different layers. "Single" indicates that each user is represented by only one representation vector. "Anti-OS" refers to the relief of impacts from over-smoothing. For Max, it is common to select a pre-defined layer.

Method	Mean	Max	Concat.	Att.
$Agg(\cdot)$	$\frac{1}{L+1}\sum_{l=0}^{L}\boldsymbol{Z}^{(l)}$	$\max_l(\boldsymbol{Z}^{(l)})$	$\Vert_{l=0}^{L}\boldsymbol{Z}^{(l)}$	$\sum_{l=0}^{L}\alpha_l\boldsymbol{Z}^{(l)}$
Time	$O(Lmd)$	$O(1)$	$O(1)$	$O(md(L+d))$
Model	LightGCN[11]	GCN[17]	NGCF[32]	MLAP[13]
Unified	✓	✗	✓	✓
Single	✓	✓	✗	✓
Anti-OS	✗	✓	✓	✓
Adaptive	✗	✗	✗	✓

2.3.3 Social Influence Exploration. Apart from explicit social influence, implicit social influence also plays an important role in social recommendations, since it aligns the social graph with the user-item bipartite graph via indirect relations between users. Different from social diversity simulation that endeavors to improve representations, social influence exploration aims to model the connectivity between users on the social network. To this end, we adopt selector layers with Gumbel sampling [1, 45], which simulates relation sampling under an extreme value distribution. This generates a new list of candidate neighbors for each user based on top-K relations. Formally, we apply K selectors to compute the scores between users. For user u and selector i, the scores between u and other users are defined as follows:

$$Sim(u, i) = act((\boldsymbol{P}_r\boldsymbol{p}_u^\top) \odot \boldsymbol{W}_{select}^{(i)}), \tag{10}$$

$$Score(u, i) = Softmax(\frac{\log(Sim(u, i) + \boldsymbol{g})}{\tau}), \tag{11}$$

where $\boldsymbol{P}_r = \frac{1}{L+1}\sum_{l=0}^{L}\boldsymbol{P}_r^{(l)}$ and $\boldsymbol{p}_u \in \boldsymbol{P}_r$ is the representation of user u in bipartite path. $\boldsymbol{W}_{select}^{(i)} \in \mathbb{R}^m$ is the learnable i-th selector. $act(\cdot)$ is the activation function. $\boldsymbol{g} \in \mathbb{R}^m$ is the Gumbel noise generated by $\boldsymbol{g} = -\log(-\log(\boldsymbol{\mu}))$, where $\boldsymbol{\mu} \sim U(0,1)$. τ is the temperature and the scores become discrete when $\tau \longrightarrow 0$. Scores from all K selectors are summed up to get the top-K candidate neighbors.

However, directly connecting users with their K candidate neighbors on the social network can damage the network architecture. If the candidate neighbors share highly similar preferences, it will harm the social influence diversity and cause user representation collapse [33]. To retain the information entropy on graph, for each user u, we perform a follow-up Bernoulli sampling on the $|\mathcal{N}(u)|$ explicit neighbors and K candidate neighbors as follows:

$$\tilde{s}_u \sim Bernoulli(\frac{|\mathcal{N}(u)|}{K + |\mathcal{N}(u)|}), \tag{12}$$

$$\widetilde{\boldsymbol{S}} = \|_{u=0}^{m}\tilde{s}_u, \tag{13}$$

where $\tilde{s}_u \in \mathbb{R}^m$ denotes the new neighbor vector of user u and $\widetilde{\boldsymbol{S}} \in \mathbb{R}^{m\times m}$ denotes the new social network. After dual sampling, the number of user u's connected neighbors should be around $|\mathcal{N}(u)|$, the initial number of neighbors. We set $\widetilde{\boldsymbol{S}} = \boldsymbol{S}$ in the first epoch of training and update it in each following epoch with the dual sampling process. As the training phases proceed, the social graph keeps being reconstructed. It could be seen as graph augmentation to activate and learn the comprehensive social pattern.

2.4 Model Optimization

Now that the user embeddings $\widetilde{\boldsymbol{P}}_s$ in the social path have gone through the social influence learning stage, they will be aggregated with the user embeddings $\boldsymbol{P}_r$ in the bipartite graph. Even though $\widetilde{\boldsymbol{P}}_s$ and $\boldsymbol{P}_r$ are both generated from the same initial embeddings, they do not exist in the same feature space because $\widetilde{\boldsymbol{P}}_s$ contains transformations from LGE-VAE. Consequently, $\widetilde{\boldsymbol{P}}_s$ may not be directly applied for item predictions on the user-item common space, i.e., the space shared by $\boldsymbol{P}_r$ and $\boldsymbol{Q}$. To obtain well-unified user embeddings, we transform $\widetilde{\boldsymbol{P}}_s$ before the aggregation:

$$\boldsymbol{P}_{final} = \boldsymbol{P}_r + act(\widetilde{\boldsymbol{P}}_s\boldsymbol{W}_{agg}), \tag{14}$$

where $\boldsymbol{W}_{agg} \in \mathbb{R}^{d\times d}$ is a learnable transformation matrix. To train the model, we use the pairwise Bayesian Personalized Ranking (BPR) [27] as our recommendation loss function:

$$\mathcal{L}_{rec} = \sum_{i\in\mathcal{I}(u), j\notin\mathcal{I}(u)} -\log\sigma(\hat{r}_{u,i}(\Omega) - \hat{r}_{u,j}(\Omega)), \tag{15}$$

where Ω refers to all learnable parameters of our model. $\hat{r}_{u,i}$ is the predicted score for user u on item i, where $\hat{r}_{u,i} = \boldsymbol{p}_u\boldsymbol{q}_i$, $\boldsymbol{p}_u \in \boldsymbol{P}_{final}$ and $\boldsymbol{q}_i \in \boldsymbol{Q}$. $\sigma(\cdot)$ is the sigmoid function. The L2 regularization is omitted for clarity. At last, we integrate the recommendation loss $\mathcal{L}_{rec}$ with the LGE-VAE loss $\mathcal{L}_{vae}$, which consists of the reconstruction loss and the KL divergence loss. Overall, the objective function of EIISRS is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \gamma\mathcal{L}_{vae}, \tag{16}$$

where γ a coefficient to rescaled $\mathcal{L}_{vae}$ such that the main recommendation task can be prioritized.

Algorithm 1 The first training epoch of EIISRS

Input: user-item interaction matrix $\boldsymbol{R}$, social networks $\boldsymbol{S}$, and initialized user embeddings $\boldsymbol{P}^{(0)}$ and item embeddings $\boldsymbol{Q}^{(0)}$.
Output: user embeddings $\boldsymbol{P}_{final}$ and item embeddings $\boldsymbol{Q}$, augmented social networks $\widetilde{\boldsymbol{S}}$
Update: Social networks $\boldsymbol{S}$ is replaced with augmented social networks $\tilde{\boldsymbol{S}}$ as input

```
for each epoch do
    for each batch do
        P_s^(0), P_r^(0) ← Eq.(1);                         ▷ Self-gating Unit
        for l = 1 : L do
            P_s^(l), P_r^(l), Q^(l) ← Eq.(2)-(4);           ▷ Graph Conv.
            μ^(l), σ^(l) ← Eq.(8);
            Z_s^(l) ← Eq.(9);                              ▷ Social Diversity Simulation
            α_l ← Eq.(5);
            Calculate the L_reconstruct and KLD;
        end for
        ~P_s = Σ_{i=0}^{L} α_l Z_s^(l);                    ▷ Social Influence Propagation
        P_final ← Eq.(14);
        Calculate the pairwise BPR loss L_rec;
    end for
    ~S ← Eq.(10)-(13);                                     ▷ Social Influence Exploration
end for
return P_final, Q, ~S
```

2.5 Complexity

For model size, the size of user and item embeddings in total is $(m + n) \times d$. Each self-gating unit and layerwise attention unit occupies $(d + 1) \times d$, altogether $3 \times (d + 1)$. K selectors take up $k \times m$ in total. Encoder occupies $2 \times d \times d$ while decoder is $d \times d$, same as the aggregation matrix. To sum up, the total size of our model is approximately equal to $(m + n + 7d + 3) \times d + k \times m$. Since $d \ll min(m, n)$ and $k \ll m$, showing our model is fairly portable.

For time complexity, regarding graph convolution, the time complexity is $O(|\boldsymbol{R}^{+}|dL)$ for the user-item bipartite graph, and $O(|\boldsymbol{S}^{+}|dL)$ for the social graph, where $|\boldsymbol{R}^{+}|$ and $|\boldsymbol{S}^{+}|$ denote the non-zero values of original graphs respectively. Compared with previous graph-based recommendation systems, we drop the feature transformation matrix and activation function, so the time complexity of graph convolution is relatively slower. For the self-gating units, layerwise attention unit, encoder/decoder of LGE-VAE, and aggregation matrix, the time complexity is as low as $O(md^2)$. As for the selector, the time complexity would b $O(|\mathcal{U}^{+}|^2 d)$ since we only choose a tiny subset of users to do the processing. Thus, our model is fairly efficient, too.

3 Experiments

3.1 Experimental Setting

3.1.1 Datasets. We use three widely-used datasets from the real world for our experiments, which are LastFM[1], Flickr[2], and Yelp[3]. Since our model is based on implicit feedback, we follow the settings

[1] http://files.grouplens.org/datasets/hetrec2011/
[2] http://flickr.com
[3] https://www.yelp.com/dataset/challenge.

of previous work [46] to binarize all ratings if needed, in which ratings less than 4 are assigned as 0 and the rest are 1. Table 2 shows the detailed statistics of the datasets.

Table 2: Statistics of datasets.

Dataset	LastFM	Flickr	Yelp
User #	1,892	8,358	17,237
Item #	17,632	82,120	38,342
Interaction #	92,834	314,809	204,448
Interaction %	0.2783	0.0459	0.0309
Relation #	25,434	187,273	143,765
Relation %	0.7105	0.2681	0.0484

3.1.2 Baselines. To verify the superiority of our model, we compare EIISRS with some representative models as shown below:

- **BPR** [27]: is one of the most popular traditional recommendation models using pairwise ranking loss. It aims to maximize the score difference between positive (beloved) items and negative (disliked) items.
- **SBPR** [48]: is an extended version of BPR that makes use of social relations to improve performance. It is based on the assumption that the positive items of users' neighbors are likely to be recommended compared with other items.
- **CDAE** [40]: is an AE-based model with the denoising mechanism for the top-K recommendation. It introduces the user embeddings in the input layers to provide collaborative signals. The reconstructed itemset is used for prediction.
- **Multi-VAE** [22]: is a VAE-based model with multinomial distribution to model user ratings and capture the collaborative signals in ratings for prediction.
- **NGCF** [32]: is one of the first models that incorporate GNNs to depict user-item interactions. It is able to capture high-order relations among the interactions through graph propagations layer by layer.
- **LightGCN** [11]: is a simplified GNN-based recommendation system. It gets rid of the reluctant transformation and nonlinear activation function.
- **DiffNet++** [37]: is the representative social recommendation model that simulates social influence diffusion based on GCN. Derived from previous work DiffNet [38], it explores a more comprehensive diffusion structure from the user's social networks to both social and item space.
- **ESRF** [45]: is a recent social recommendation model that enhances the effect of social networks using adversarial learning. Alternative neighbors of each user would be generated and assigned an attentive score for aggregation and distinguishing their usefulness.

3.1.3 Evaluation Metrics. For recommendation evaluation, we adopt four widely used metrics for our task: classification-based metrics *Precision@k*, *Recall@k*, *F1@k* and a ranking-based metric *NDCG@k*. We perform item ranking on all items rather than sampled item sets to ensure that the evaluation is unbiased and robust. Improvements over 1% are considered significant [29].

3.1.4 Settings. For a fair comparison, we assign the best parameter settings for each baseline method as previous works did. The proposed model EIISRS adopts the Adam optimizer for all models and uses grid search to fine-tune all hyperparameters. For general settings, the dimension of latent representation is fixed at 50. The initial learning rate is $1e^{-3}$ and the batch size is 2000 for top-10 recommendations. The coefficients are 0.1, 0.01, and 0.2 for reconstruction loss, KL divergence, and temperature of Gumbel sampling, respectively. The number of layers of GCN is 2, and the number of candidate neighbors is 30, which we will discuss in detail in the following hyperparameter analysis section. For precise assessment, we split the dataset into the training set and the test set with a proportion of 8:2. We associate one positive item with one negative item in each training sample.

3.2 Experimental Results

3.2.1 Overall Performance. The overall performance results of all the models are shown in Table 3. We highlight the best model in boldface and underline the follow-up model. The improvements of our model compared with the follow-up model are also listed at the bottom of the table. By analyzing the results, we can draw the following conclusions:

- We categorize all the models to enable better comparisons among them. Accordingly, we have the traditional statistical models (i.e., BPR, SBPR), AE-based models (i.e., CDAE, Multi-VAE), graph-based general recommendation models (i.e., NGCF, LightGCN), and graph-based social recommendation models (i.e., DiffNet++, ESRF). We observe that the graph-based models outperform non-graph-based models since applying GNN message-passing on the social network can help improve generalization. To our surprise, AE-based models have the poorest performance among all the baselines. This should be attributed to the naive structure of models and their inability in capturing implicit feedback.
- With the social network as available side information, we observe that social recommendation models (i.e., SBPR, DiffNet++, ESRF) outperform general recommendation models (i.e., BPR, NGCF) in most cases. This demonstrates the fact that social influence contains much useful information for improving recommendations. We can take advantage of such resources if we properly manage the data. It is worth noting that LightGCN has a great capacity to exploit the bipartite graph for better recommendations, and thus, it outperforms some social recommendation models on certain datasets. Our model incorporates LightGCN to assist implicit social influence exploration. As a result, the effectiveness of social graph imputation can be well enhanced.
- Overall, EIISRS significantly outperforms all other baselines. It is convinced that EIISRS can disentangle the characteristics of social influence and adaptively learn comprehensive social patterns from both the social graph and the bipartite graph. Referring to the dataset statistics, the percentage of social links varies among different datasets. This implies that exploring useful social information can be a challenging task. We observe that when the social graph is sparse (i.e.,

Table 3: Overall recommendation performance comparison.

Model	LastFM				Flickr				Yelp			
	P@10	R@10	F1@10	N@10	P@10	R@10	F1@10	N@10	P@10	R@10	F1@10	N@10
BPR	0.1157	0.1180	0.1168	0.1452	0.0019	0.0020	0.0019	0.0021	0.0019	0.0071	0.0030	0.0045
SBPR	0.1559	0.1564	0.1561	0.2019	0.0018	0.0018	0.0013	0.0024	0.0032	0.0121	0.0051	0.0074
CDAE	0.0364	0.0755	0.0491	0.0682	0.0013	0.0034	0.0019	0.0026	0.0013	0.0110	0.0023	0.0054
Multi-VAE	0.0950	0.1825	0.1250	0.1607	0.0015	0.0044	0.0022	0.0031	0.0028	0.0232	0.0050	0.0118
NGCF	0.1662	0.1708	0.1685	0.2079	0.0026	0.0034	0.0030	0.0034	0.0041	0.0162	0.0066	0.0098
LightGCN	0.1631	0.1676	0.1653	0.2137	<u>0.0033</u>	0.0039	0.0036	0.0044	<u>0.0061</u>	<u>0.0238</u>	<u>0.0097</u>	<u>0.0149</u>
DiffNet++	0.1722	0.1751	0.1736	0.2069	0.0030	0.0032	0.0031	0.0038	0.0049	0.0179	0.0076	0.0111
ESRF	<u>0.1913</u>	<u>0.1968</u>	<u>0.1940</u>	<u>0.2465</u>	<u>0.0033</u>	<u>0.0046</u>	<u>0.0039</u>	<u>0.0047</u>	0.0055	0.0209	0.0088	0.0130
EIISRS	**0.1953**	**0.2004**	**0.1978**	**0.2532**	**0.0036**	**0.0047**	**0.0041**	**0.0051**	**0.0066**	**0.0244**	**0.0103**	**0.0153**
Improv.	**2.1%**	**1.8%**	**2.0%**	**2.7%**	**9.1%**	**2.2%**	**5.1%**	**7.8%**	**8.2%**	**2.5%**	**6.2%**	**2.7%**

Flickr, Yelp), the performance of graph-based general recommendation models even surpasses that of graph-based social recommendation models. Still, under our design, EIISRS is robust and holds the lead when datasets are of high sparsity.

3.2.2 *Ablation Study.* To investigate the effectiveness of each component in the model, we create four variants by removing certain parts of EIISRS, and we compare them with our original complete model (i.e., EIISRS). SIP, SID, and SIE refer to the variants where Social Influence Propagation, Social Influence Diversity, and Social Influence Exploration have been removed from the model, respectively. Vanilla refers to the backbone model, which has removed all three components. From Figure 3, we can easily observe that the performance becomes worse after removing any components. The most significant performance drop happens in SID. Therefore, we are convinced that the simulation of social influence diversity is well-suited to tackle the sparsity and inconsistency issue on the social network for social recommendations. Meanwhile, by comparing with Vanilla, we conclude that every component is contributing to our model, and the corporations between components matter.

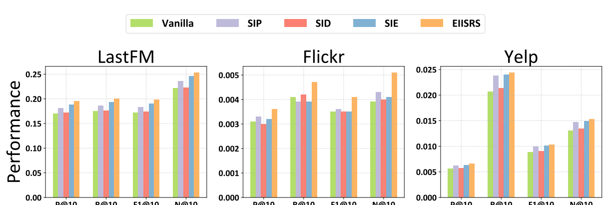

Figure 3: Ablation study on different components.

In LightGCN, the authors demonstrated that the weighted sum of different layers is redundant in their model. To verify the validity of layerwise attention in our case, we set up another ablation study on the aggregation methods. As shown in Figure 4, we may find that Attention-based aggregation outperforms any other aggregation methods, including Mean pooling, Max pooling, and Concatenation. The reasons are as follows: (1) in the LightGCN case, layerwise attention is ineffective due to the user-item alternation on the bipartite graph, but in our case, it can well capture the patterns of social influence propagation on the social graph; (2) by combing the attentive aggregation with social influence diversity, the social information from different scopes of neighborhoods can be further generalized and disentangled.

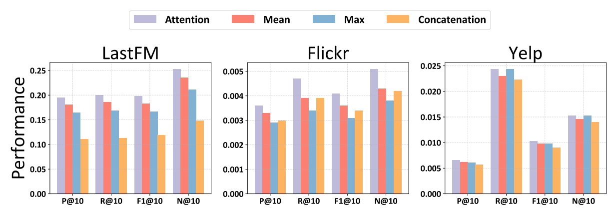

Figure 4: Ablation study on aggregation methods.

In addition, we investigate the influence of different sampling methods in exploring the implicit neighborhoods, which has not been done in the previous work [45]. We choose two more sampling methods that are commonly used (i.e., Gaussian, Uniform), along with the Gumbel and None as the control groups. As shown in Table 4, we notice that Gumbel sampling methods can always achieve the best performance in both ESRF and our model, while Gaussian and Uniform sampling methods run neck-to-neck. The reasons for the effectiveness of Gumbel sampling are: 1) the concrete distribution with the Gumbel-max trick enhances stability under max operation (e.g., top-k selection), and 2) it can represent any discrete distribution by discretization [25]. Another observation is that without any sampling applied, the performance deterioration of EIISRS is more significant than that of ESRF. This is due to the different downstream processing between the two models, where ESRF tries to attentively aggregate the implicit neighbors' representations, while EIISRS tries to build the new augmented social graph for the next training. As a result, EIISRS is more dependent on a proper sampling method to avoid noise inference before the Bernoulli sampling.

3.2.3 *Hyperparamter Analysis.* Figure 5 shows the model performance with different numbers of generated implicit neighbors. We observe that the model performance reaches its peak when there are 30 new neighbors. Moreover, even when the number of neighbors becomes large enough, the curve fluctuates only within a small

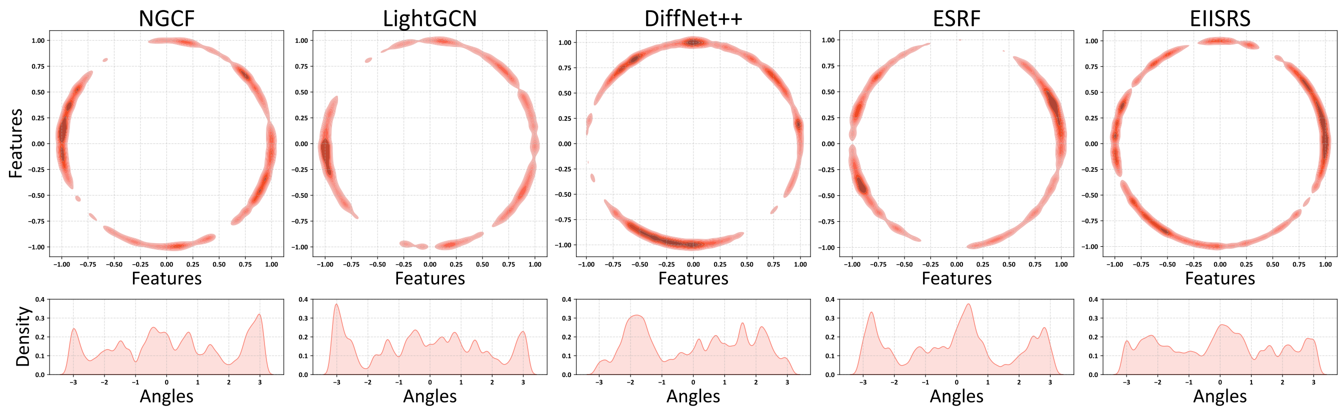

Figure 7: Visualization of user representations on LastFM.

Table 4: Ablation study on different sampling methods for ESRF and EIISRS on LastFM.

Model	Method	P@10	R@10	F1@10	N@10
ESRF	**Gumbel**	**0.1913**	**0.1968**	**0.1940**	**0.2465**
	Gaussian	0.1902	0.1954	0.1927	0.2447
	Uniform	0.1900	0.1948	0.1923	0.2449
	None	0.1895	0.1946	0.1920	0.2430
EIISRS	**Gumbel**	**0.1953**	**0.2004**	**0.1978**	**0.2532**
	Gaussian	0.1921	0.1969	0.1944	0.2504
	Uniform	0.1921	0.1977	0.1948	0.2468
	None	0.1883	0.1945	0.1913	0.2439

range, and the model still outperforms most other representative models. This reflects that the common over-sensitivity issue of the hyperparameter has been well alleviated, which is attributed to the effectiveness of the Bernoulli Sampling in our model, such that the social network structure is not damaged even with a large number of implicit candidate neighbors.

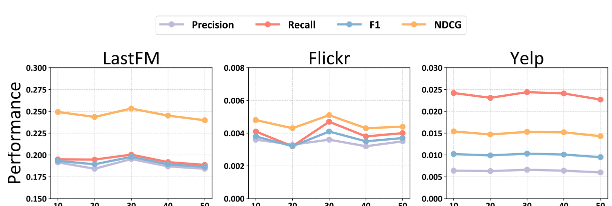

Figure 5: Impact of the number of generated neighbors.

In addition, Figure 6 shows the model performance with different temperatures in Gumbel sampling. We can also observe that our model reaches its best performance when τ is around 0.2. Similar to the outputs of different generated neighbors, there are few fluctuations between different results. We are convinced that not only the number of generated neighbors but also the temperature benefit from the design of the dual sampling process.

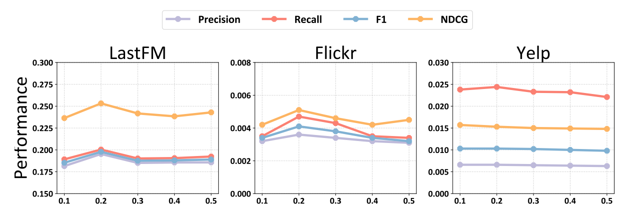

Figure 6: Impact of temperature in Gumbel sampling.

3.3 Case Study

This section exhibits some case studies of our model as supplements, in order to illustrate the validity of our hypothesis.

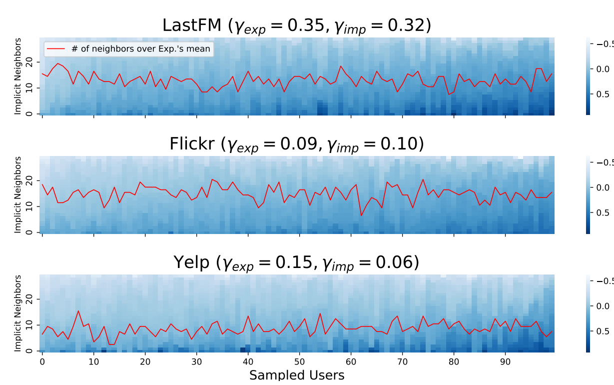

Figure 8: Visualization of the similarity between sampled users and their implicit neighbors generated from SIE. The red line shows the number of implicit neighbors having higher similarity than that of average explicit neighbors.

3.3.1 Anlaysis of User Representation Collapse. User representation collapse is defined as the phenomenon where users are stuck in the observable items but ignore the huge potential of diverse interests [33]. Under the design of EIISRS, two novel components (i.e., Social Diversity Simulation, Dual Sampling Process) can alleviate this problem effectively. We follow the previous work [47] to analyze user representation collapse with visualization on LastFM for simplicity. We map the final user representation generated by different models into two-dimensional normalized vectors using t-SNE [30], and we plot them with Gaussian Kernel Density Estimation (KDE). The upper figure shows the distribution of user representation and the bottom one is plotted based on their angles (i.e., $arctan2(y, x)$). Note that NGCF and LightGCN are general recommendation models, while DiffNet++, ESRF, and EIISRS are social recommendation models. According to Figure 7, we can observe that our model yields a more rounded circular graph on the top and smoother curve peaks on the bottom graph compared with other models. It demonstrates that EIISRS can well relieve the user representation collapse problem when it comes to a sparse user-item interaction graph or social graph.

3.3.2 Analysis of Implicit Social Graph. In Figure 8, the heatmap shows the similarity between sampled users and their generated implicit neighbors based on user embeddings, while γ_{exp} and γ_{imp} denote users' average similarities with their explicit and implicit neighbors computed based on the embeddings of their ground-truth purchased items. We can observe that all kinds (i.e., exp. and imp.) of neighbors have low similarities, which reflects the diversity of social influence. Besides, according to the similarity, EIISRS can learn useful implicit social influence on Flickr that better represents users' behavior and results in the greatest improvement. On LastFM and Yelp, the implicit social influence rather serves as a regularization, which contributes more to the sparser social graph (i.e., Yelp) and is suppressed on the informative one (i.e., LastFM). This illustrates two practical ways to explore the potential of social influence in future work given the different natures of social graphs.

4 Related Work

4.1 Social Recommendation Systems

Social recommendation systems have been around since the early days of social networking in the 2000s [12], and have developed rapidly in the past decade. Social recommenders are based on the premise that individuals trust the opinions of their socially connected neighbors more than anonymous sources, rather than focus solely on the user's historical behavior. SoRec [23] is one of the earliest papers on social recommendations in the manner of machine learning. They offer the formal definitions of social recommendation and analyze its distinguishing features and implications in contrast to general recommendations. SocialMF [14] and TrustSVD [8] follow and extend the previous work [18, 19] to introduce the social impacts into recommendations. They employ social trusts to transmit the preference and lessen the data sparsity and cold start concerns. However, such prior methods that can be classified as social regularization [36] are unable to extract the high-order social relations and apply them to recommendations. With the development of the graph neural network (GNN) [7, 10, 17], it has been frequently applied to explore the comprehensive patterns among graphs [5, 11, 32]. By combining the bipartite graph with a social graph, not only are the user representations learned from the item domain but also the social relations. DiffNet++ [37] applies the influence and interest diffusion on different graphs and attentively aggregates them. ESRF [45] exploits motif-based hypergraphs to learn the enhanced social recommendations with adversary learning. Recently, self-supervised learning also empowers great capability for recommendations [35, 44, 46, 47]. Nevertheless, none of them have elaborated on the mechanism of social influence through disentanglement in social recommendations.

4.2 Variational Autoencoder for Recommendation Systems

Variational Autoencoder (VAE) [15] is a deep generative model that has gained much appeal in the field of machine learning given its capacity to learn a low-dimensional representation of input data. The training of VAE entails maximizing the lower bound on the data's probability, which is stated as the sum of the reconstruction loss from latent representations and Kullback-Leibler (KL) divergence with a standard normal distribution. Many works on recommendations also employ this powerful technique [20, 22, 24, 28, 34, 49], where the users' preferences are encoded into embeddings with the objective of reconstructing historical ratings. Later, the Variational Graph Autoencoder (VGAE) [16] follows the idea of VAE and utilizes graph-structured data as input. Specifically, the encoder of VGAE is usually a GCN model that is more expressive than the traditional encoder (i.g., MLP). In this way, VGAE-based link prediction models [4, 6, 9] show their superiority in learning the graph structure and retrieving better performance. MS-VGAE [9] learns the multi-scale information by retrieving the latent embeddings in different dimensions. VDGAE [6] applies disentanglement by setting multi-head frameworks and minimizing the mutual information between each of the two channels in case of intertwinement. Besides, HVGAE [2] and ReLearn [41] reveal the potential of using VGAE on recommendation systems and social network learning, respectively. However, all of these methods rely heavily on node/edge features, and whether VGAE works on general graph recommendations and social recommendations is still underexplored.

5 Conclusion

In our work, we develop a novel recommendation model that mainly focuses on tackling the incompatibility problem inherent in the social network. We learn the layerwise representations of neighborhoods on the social graph and design LGE-VAE to simulate the social influence diversity. Then, we apply a layerwise graph attention network to capture the propagation of social influence. Finally, we propose a new dual sampling method to explore implicit social neighbors. With dual sampling, the representation collapse problem caused by user sparsity can be well addressed. Extensive empirical studies provide solid evidence that our model is feasible, efficient, and agile in design. In addition, specific user case studies have been conducted to investigate the potential usage of social networks for recommendations in the future.

Acknowledgments

This work was supported by the Science and Technology Development Fund, Macau SAR (0068/2020/AGJ, 001/2024/SKL), GDST (2019B090909005, 2020B1212030003, 2023A0505030013), Nansha District (2023ZD001), MYRG2022-00192-FST, MYRG-GRG2023-00186-FST-UMDF.

References

[1] Abubakar Abid, Muhammed Fatih Balin, and James Zou. 2019. Concrete Autoencoders for Differentiable Feature Selection and Reconstruction. *arXiv preprint arXiv:1901.09346* (2019).

[2] Basmah Altaf, Uchenna Akujuobi, Lu Yu, and Xiangliang Zhang. 2019. Dataset Recommendation via Variational Graph Autoencoder. *2019 IEEE International Conference on Data Mining (ICDM)* (2019), 11–20.

[3] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. 933–941.

[4] Haoyi Fan, Fengbin Zhang, Yuxuan Wei, Zuoyong Li, Changqing Zou, Yue Gao, and Qionghai Dai. 2022. Heterogeneous Hypergraph Variational Autoencoder for Link Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 8 (2022), 4125–4138.

[5] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. New York, NY, USA, 417–426.

[6] Jun Fu, Xiaojuan Zhang, Shuang Li, and Dali Chen. 2023. Variational Disentangled Graph Auto-Encoders for Link Prediction. arXiv:2306.11315 [cs.LG]

[7] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. 1263–1272.

[8] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. 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, Texas) *(AAAI'15)*. 123–129.

[9] Zhihao Guo, Feng Wang, Kaixuan Yao, Jiye Liang, and Zhiqiang Wang. 2022. Multi-Scale Variational Graph AutoEncoder for Link Prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. New York, NY, USA, 334–342.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, Vol. 30.

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. 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* (Virtual Event, China) *(SIGIR '20)*. New York, NY, USA, 639–648.

[12] Tad Hogg and Lada Adamic. 2004. Enhancing Reputation Mechanisms via Online Social Networks. In *Proceedings of the 5th ACM Conference on Electronic Commerce* (New York, NY, USA) *(EC '04)*. New York, NY, USA, 236–237.

[13] Takeshi D. Itoh, Takatomi Kubo, and Kazushi Ikeda. 2022. Multi-level attention pooling for graph neural networks: Unifying graph representations with multiple localities. *Neural Networks* 145 (2022), 356–373.

[14] Mohsen Jamali and Martin Ester. 2010. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (Barcelona, Spain) *(RecSys '10)*. New York, NY, USA, 135–142.

[15] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[16] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *arXiv:1611.07308 [cs, stat]* (2016).

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[18] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) *(KDD '08)*. New York, NY, USA, 426–434.

[19] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[20] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. New York, NY, USA, 305–314.

[21] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated Graph Sequence Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[22] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. Republic and Canton of Geneva, CHE, 689–698.

[23] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (Napa Valley, California, USA) *(CIKM '08)*. New York, NY, USA, 931–940.

[24] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. *Learning Disentangled Representations for Recommendation*. Red Hook, NY, USA.

[25] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *ArXiv* abs/1611.00712 (2016).

[26] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology* 27 (2001), 415–444.

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. Arlington, Virginia, USA, 452–461.

[28] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. New York, NY, USA, 528–536.

[29] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A Review-Aware Graph Contrastive Learning Framework for Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. New York, NY, USA, 1283–1293.

[30] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[32] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. New York, NY, USA, 165–174.

[33] Bingchao Wu, Yangyuxuan Kang, Bei Guan, and Yongji Wang. 2023. We Are Not So Similar: Alleviating User Representation Collapse in Social Recommendation. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval* (Thessaloniki, Greece) *(ICMR '23)*. New York, NY, USA, 378–387.

[34] Ga Wu, Mohamed Reda Bouadjenek, and Scott Sanner. 2019. One-Class Collaborative Filtering with the Queryable Variational Autoencoder. 921–924.

[35] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-Supervised Graph Learning for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. New York, NY, USA, 726–735.

[36] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2023. A Survey on Accuracy-Oriented Neural Recommendation: From Collaborative Filtering to Information-Rich Recommendation. *IEEE Trans. on Knowl. and Data Eng.* 35, 5 (2023), 4425–4445.

[37] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2022. DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (2022), 4753–4766.

[38] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 235–244.

[39] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual Graph Attention Networks for Deep Latent Representation of Multifaceted Social Effects in Recommender Systems. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. New York, NY, USA, 2091–2102.

[40] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, California, USA) *(WSDM '16)*. New York, NY, USA, 153–162.

[41] Carl Yang, Jieyu Zhang, Haonan Wang, Sha Li, Myungwan Kim, Matt Walker, Yiou Xiao, and Jiawei Han. 2020. Relation Learning on Social Networks with Multi-Modal Graph Edge Variational Autoencoders. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. New York, NY, USA, 699–707.

[42] Junliang Yu, Min Gao, Jundong Li, Hongzhi Yin, and Huan Liu. 2018. Adaptive Implicit Friends Identification over Heterogeneous Network for Social Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. New York, NY, USA, 357–366.

[43] Junliang Yu, Min Gao, Wenge Rong, Wentao Li, Qingyu Xiong, and Junhao Wen. 2017. Hybrid attacks on model-based social recommender systems. *Physica A:*

Statistical Mechanics and its Applications 483 (2017), 171–181.

[44] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. New York, NY, USA, 2084–2092.

[45] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui. 2022. Enhancing Social Recommendation With Adversarial Graph Convolutional Networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 08 (2022), 3727–3739.

[46] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. New York, NY, USA, 413–424.

[47] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. New York, NY, USA, 1294–1303.

[48] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (Shanghai, China) *(CIKM '14)*. New York, NY, USA, 261–270.

[49] Yaochen Zhu and Zhenzhong Chen. 2022. Mutually-Regularized Dual Collaborative Variational Auto-Encoder for Recommendation Systems. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. New York, NY, USA, 2379–2387.