

# Robust Preference-Guided Denoising for Graph based Social Recommendation

Yuhan Quan  
Department of Electronic  
Engineering, Tsinghua University  
Beijing, China

Jingtao Ding\*  
Department of Electronic  
Engineering, Tsinghua University  
Beijing, China

Chen Gao  
Department of Electronic  
Engineering, Tsinghua University  
Beijing, China

Lingling Yi  
Wechat, Tencent  
Shenzhen, China

Depeng Jin  
Department of Electronic  
Engineering, Tsinghua University  
Beijing, China

Yong Li  
Department of Electronic  
Engineering, Tsinghua University  
Beijing, China

## ABSTRACT

Graph Neural Network (GNN) based social recommendation models improve the prediction accuracy of user preference by leveraging GNN in exploiting preference similarity contained in social relations. However, in terms of both effectiveness and efficiency of recommendation, a large portion of social relations can be redundant or even noisy, *e.g.*, it is quite normal that friends share no preference in a certain domain. Existing models do not fully solve this problem of relation redundancy and noise, as they directly characterize social influence over the full social network. In this paper, we instead propose to improve graph based social recommendation by only retaining the informative social relations to ensure an efficient and effective influence diffusion, *i.e.*, graph denoising. Our designed denoising method is preference-guided to model social relation confidence and benefits user preference learning in return by providing a denoised but more informative social graph for recommendation models. Moreover, to avoid interference of noisy social relations, it designs a self-correcting curriculum learning module and an adaptive denoising strategy, both favoring highly-confident samples. Experimental results on three public datasets demonstrate its consistent capability of improving two state-of-the-art social recommendation models by robustly removing 10-40% of original relations. We release the source code at <https://github.com/tsinghua-fib-lab/Graph-Denoising-SocialRec>.

## KEYWORDS

Social Recommendation, Graph Denoising, Preference Learning

### ACM Reference Format:

Yuhan Quan, Jingtao Ding, Chen Gao, Lingling Yi, Depeng Jin, and Yong Li. 2023. Robust Preference-Guided Denoising for Graph based Social Recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW '23, May 1–5, 2023, Austin, TX, USA)*

\*Corresponding author (dingjt15@tsinghua.org.cn)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

WWW '23, May 1–5, 2023, Austin, TX, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-9416-1/23/04...\$15.00  
<https://doi.org/10.1145/3543507.3583374>

'23), May 1–5, 2023, Austin, TX, USA. ACM, New York, NY, USA, 12 pages.  
<https://doi.org/10.1145/3543507.3583374>

## 1 INTRODUCTION

Nowadays, social recommendation has become an important scenario of personalized recommender systems, where both user-item interactions and user-user social connections are available in the platforms [40]. Previous studies on human behaviors have established two basic theories describing the impact of social relations between users: 1) socially-connected users tend to possess similar preferences, which is referred as *social homophily* [33], and 2) behavior of a user can be influenced by her friends, like co-purchasing the same item, which is referred as *social influence* [32]. Above theories regarding the impacts of social relations on recommendation have been validated by recent applications in areas like advertising [2], e-commerce [6, 54] and etc.

To characterize the above social relation effects on user preference, researchers have developed many social recommendation methods, like achieving preference similarity with social regularization [31], enhancing influence from trust relations [18], and exploiting different relations with strong or weak ties [46]. Recent progress of graph neural network (GNN) in graph machine learning (ML) has further boosted the development of graph based social recommendation [15], as messaging passing mechanism widely adopted in GNN is well suited for characterizing influence diffusion in social context [13, 41, 51]. Moreover, aided by self-supervised learning techniques, GNN based social recommendation (short for GSocRec) models can extract more informative signals from massive observed social relations [58, 60].

However, contrastive to the prevalence of GSocRec models, one important problem that has been barely explored is the existence of redundant or even noisy social relations in the context of graph based social recommendation. As we have shown in Figure 1(a), most users only share similar preferences with a small portion of friends. Specifically, the median ratio value of friends having co-interactions is about 30% and 20% in two empirical datasets, respectively. Therefore, the rest social relations may be redundant or even harmful for social recommendation, as it is quite possible that two friends have distinct preferences in some areas. Consequently, previous GSocRec models that directly characterize social influence over the full social network have two shortcomings. One is the huge burden on both computation and storage brought by learning

GNN on the full user-user graph, as the size of social relations can easily approach to hundreds of billion in current commercial social recommendation platforms like Facebook and Wechat [28, 36] (both having over billions of users). This efficiency issue further hinders the graph structure learning based approaches [49, 59] that aim to learn a more informative graph by operating on the cartesian product space of nodes. Another is the possible negative impact on recommendation accuracy. Although some previous works propose to deal with this diversity of social influence via attention mechanism [50, 55] or expectation-maximization method [47], they still lack groundtruth labels and thus cannot learn the degree of social influence effectively.

To fully resolve the above problem, we propose to improve graph based social recommendation by denoising social network, *i.e.*, removing those redundant or noisy social relations from the original graph and only retaining the informative ones for much more efficient and effective learning of GSocRec models. However, graph denoising for GSocRec models is non-trivial due to the following two challenges:

- How to overcome the difficulty of identifying the informative social relations for recommendation purpose? Reasons behind users' social relation formation are complicated, which do not necessarily induce similar user preferences, increasing the difficulty of identifying useful relations.
- How to achieve robust graph denoising given the existence of noisy labels? In the context of social recommendation, a large portion of user relations are useless or even harmful for enhancing user preference learning. Thus they can be seen as noisy signals and pose robustness requirements on graph denoising.

In this paper, we propose a novel denoising-enhanced recommendation framework for GSocRec models as illustrated in Figure 1(b). The core of this framework is a **Graph Denoising Method for Social Recommendation** (shorten as GDMSR), which leverages user preferences to identify informative social relations from massive candidates and retain them as the denoised social graph, empowering the downstream GSocRec model with an arbitrary structure. To solve the first challenge of characterizing relation confidence, GDMSR is designed to better exploit preference guidance in a two-fold manner, *i.e.*, explicit preference-based relation modeling and implicit co-supervision by a recommendation loss. As for the second challenge of robust denoising, it is equipped with a self-correcting curriculum learning module and an adaptive denoising strategy that both favor highly-confident samples (*i.e.*, useful social relations for preference learning). To summarize, our main contributions are as follows.

- We rethink the reliability of social network in context of graph based social recommendation and provide a novel angle of improving by graph denoising.
- We design a graph denoising framework that is preference-guided to model social relation confidence and benefits user preference learning in return, compatible with general GSocRec models.
- Experiment results on three public datasets demonstrate the superiority of our GDMSR over state-of-the-art baselines in terms of empowering a series of GSocRec models. Ablation studies on both empirical and synthetic datasets further support the rationality behind specific method design. Moreover, aided by

a robust preference-guided graph denoising capability, GDMSR can achieve 10-40% of reduction ratio on social graphs without hurting recommendation accuracy, as well as maintaining high practicality in terms of efficiency and scalability.

## 2 PRELIMINARIES

### 2.1 GNN based Social Recommendation Model

Generally, a typical GNN based social recommendation (short for GSocRec) model involves two key parts of social influence diffusion and user preference learning. Specifically, the former characterizes the impact of friends' preferences on a certain user, while the latter stands for training a prediction model based on user-item interactions, as a common step in recommender systems. Mathematically, a social network of  $N$  users is denoted as a user-user graph  $\mathcal{G}_s(\mathcal{U}, \mathcal{E})$ , where  $\mathcal{U} = \{u\}$  and  $\mathcal{R} = \{(u, v) | r_{uv} = 1, \forall u, v \in \mathcal{U}\}$  denote the set of users and social relations, respectively. Besides, there is another bipartite graph  $\mathcal{G}_r(\{\mathcal{U} \cup \mathcal{I}\}, \mathcal{P})$  that stands for the interactions  $\mathcal{P}$  between  $N$  users  $\mathcal{U}$  and  $M$  items  $\mathcal{I}$ . To represent users and items, GSocRec models normally use  $F$ -dimensional latent embedding matrices, *i.e.*,  $\mathbf{E}_1 \in \mathbb{R}^{N \times D}$  and  $\mathbf{E}_2 \in \mathbb{R}^{M \times D}$ , that can be learned from one-hot encoding of ids (users or items), continuous feature values, discrete feature ids, or their concatenations.

To disentangle social influence and user's own preferences, GSocRec models further use two different user preference representations, *i.e.*,  $\mathbf{E}_{1,s}$  and  $\mathbf{E}_{1,r}$ , to characterize above two types of effects, respectively. For social influence diffusion, GSocRec models leverage the message-passing mechanism of GNN to model the influence diffusion of a user  $u$ 's friends, denoted as  $\mathcal{R}_u = \{v | r_{uv} = 1\}$ , on  $u$ 's preference. Specifically, the  $u$ 's preference representation after  $K$ -hop social influence propagation is

$$\mathbf{E}_{1,s}^{(K)}(u) = \text{GNN} \left( \mathbf{E}_1^{(K-1)}(u), \left\{ \mathbf{E}_1^{(K-1)}(v) | \forall v \in \mathcal{R}_u \right\} \right), \quad (1)$$

where the GNN module can be an arbitrary GNN model such as GCN [23], GAT [44], GraphSAGE [19] and etc. On the other hand, given  $u$ 's interaction history  $\mathcal{P}_u = \{i | p_{ui} = 1\}$ , *i.e.*, the set of items that  $u$  has interacted before, the preference representation of  $u$  after  $K$ -hop preference propagation is

$$\mathbf{E}_{1,r}^{(K)}(u) = \text{GNN} \left( \mathbf{E}_1^{(K-1)}(u), \left\{ \mathbf{E}_2^{(K-1)}(i) | \forall i \in \mathcal{P}_u \right\} \right), \quad (2)$$

where another GNN module is used to update  $\mathbf{E}_{1,r}^{(K)}$ . The comprehensive representation of user preferences after considering both social influence diffusion and user preference learning is

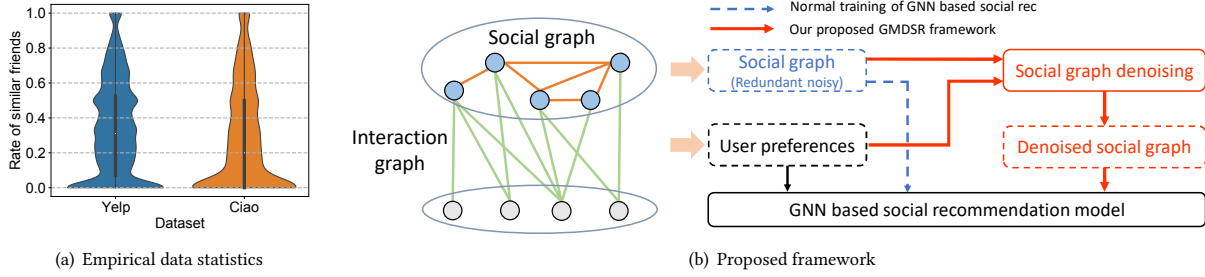
$$\mathbf{E}_1^{(K)}(u) = \text{Combine} \left( \mathbf{E}_{1,s}^{(K)}(u), \mathbf{E}_{1,r}^{(K)}(u) \right), \quad (3)$$

where  $E_1^{(0)}(u) = E_1(u)$  and the Combine module can be either a mean pooling operation [13, 41] or an attentive aggregation operation [50]. As for item representations, it is also calculated in a similar message-passing manner as

$$\mathbf{E}_2^{(K)}(i) = \text{GNN} \left( \mathbf{E}_2^{(K-1)}(i), \left\{ \mathbf{E}_1^{(K-1)}(u) | \forall u \in \mathcal{P}_i \right\} \right), \quad (4)$$

which aggregates information propagated from item  $i$ 's neighbors  $\mathcal{P}_i = \{u | p_{ui} = 1\}$ , *i.e.*, the set of users that have interacted with  $i$ .

Then, based on above  $2 \times (K + 1)$  representations of  $u$  and  $i$ , the prediction score of  $u$ 's preference over  $i$ , denoted as  $\hat{p}_{ui}$ , is calculated as



**Figure 1: (a) Distribution plot w.r.t. ratio of friends having co-interactions. (b) Our proposed denoising enhanced social recommendation framework.**

$$\hat{p}_{ui} = \mathbf{E}_1^*(u) \cdot \mathbf{E}_2^*(i),$$

$$\text{where } \mathbf{E}_1^*(u) = \frac{\sum_{k=0}^K \mathbf{E}_1^{(k)}(u)}{K+1}, \mathbf{E}_2^*(i) = \frac{\sum_{k=0}^K \mathbf{E}_2^{(k)}(i)}{K+1}. \quad (5)$$

To train a recommendation model, a widely-adopted approach is to minimize following Bayesian Personalized Ranking (BPR) loss function [34]:

$$\mathcal{L}^{BPR} = \sum_{(u,i,j) \in \bar{\mathcal{P}}} -\ln \sigma(\hat{p}_{ui} - \hat{p}_{uj}), \quad (6)$$

where  $\sigma$  is a sigmoid activation function and  $\bar{\mathcal{P}} = \{(u, i, j) | (u, i) \in \mathcal{P}, (u, j) \notin \mathcal{P}\}$  contains training triples  $(u, i, j)$  made up of an observed user-item interaction and another unobserved one. Learning above BPR objective is equivalent to maximizing the likelihood of observing such pairwise ranking relations  $\hat{p}_{ui} > \hat{p}_{uj}$ .

## 2.2 Discussion on Shortcomings

Previous works on GSocRec models generally follow the above paradigm, which focuses on enhancing user preference learning by leveraging the effect of social influence. Therefore, the vital part of GSocRec models is the modeling of social influence diffusion that aims to identify friends with similar interests as the target user and propagate their influence to this user effectively. To achieve this, GSocRec models characterize diversity of social influence by an attention mechanism in message passing, *i.e.*, using GAT in Equation (1) [50, 53, 55]. However, all these methods have to learn a recommendation model on a full  $\mathcal{G}_s$ , meaning that massive but useless message-passing operations are conducted between those weakly-connected users, which causes a severe efficiency issue. Moreover, since social relations between users are formed because of various reasons, it is quite possible that  $u$  and  $v$  share no interest or have contrast interests in a certain domain. Thus retaining  $(u, v)$  can even degrade recommendation accuracy, as it introduces noisy information. In this circumstance, besides the redundancy issue, the noise problem should also be tackled in GSocRec models.

## 3 PROPOSED METHOD

To resolve the problems brought by redundant and noisy social relations that have hindered effective learning of social recommendation models, we propose the GMDSR method that can serve as a general framework for enhancing an arbitrary GNN based social recommendation model by denoising the original social graph. On

the one hand, GMDSR leverages user preference signals to guide the graph denoising process, which helps identify informative social relations in the context of social recommendation. On the other hand, to achieve robust learning on the original social graph with noisy relations, GMDSR further incorporates a self-correcting curriculum learning mechanism that is less impacted by these noisy labeled data. After the above effective and robust graph denoising training process, the obtained model is used to denoise the social network in an adaptive manner.

### 3.1 Denoising-enhanced Social Recommendation Framework

Instead of directly training a social recommendation model given the whole social network  $\mathcal{G}_s$ , our proposed GMDSR first learns to denoise  $\mathcal{G}_s$  and obtain a much smaller but more informative graph  $\mathcal{G}_s^d$ . Then a GNN based social recommendation model utilizes  $\mathcal{G}_s^d$  as its social graph to enhance user preference learning from the interaction graph  $\mathcal{G}_r$ .

**Denoising user relations for social recommendation.** Given an original user-connected graph that contains redundant or noisy user relations (*i.e.*, edges), the denoising task is defined as a link prediction like task [27] that aims to generate a confidence score  $\hat{r}_{uv}$  of link formation for each user pair  $(u, v)$ . Differently, only those observed user relations are checked in the denoising process by ranking according to  $\{\hat{r}_{uv}\}$ . Finally, the denoised graph  $\mathcal{G}_s^d$  is obtained by removing those  $(u, v)$ s with a rather lower  $\hat{r}_{uv}$ .

**Training a social recommendation model with the denoised graph.** Given  $\mathcal{G}_s^d$ , as well as  $\mathcal{G}_r$ , any social recommendation model can be incorporated into GMDSR with no limitation on its structure. Moreover, since two training stages of denoising and recommendation are decoupled, one specific  $\mathcal{G}_s^d$  can cooperate with multiple types of recommendation models in a unified way.

Since GMDSR is generally workable for all GNN based social recommendation models discussed in Sec. 2, we focus on introducing the design of graph denoising network in the rest of this section, which has also been illustrated in Figure 2.

### 3.2 Preference-guided Graph Denoising Network

A key challenge of denoising user-connected social network for recommendation is the lack of groundtruth labels on redundant social relations, and thus every existed link is equally treated when

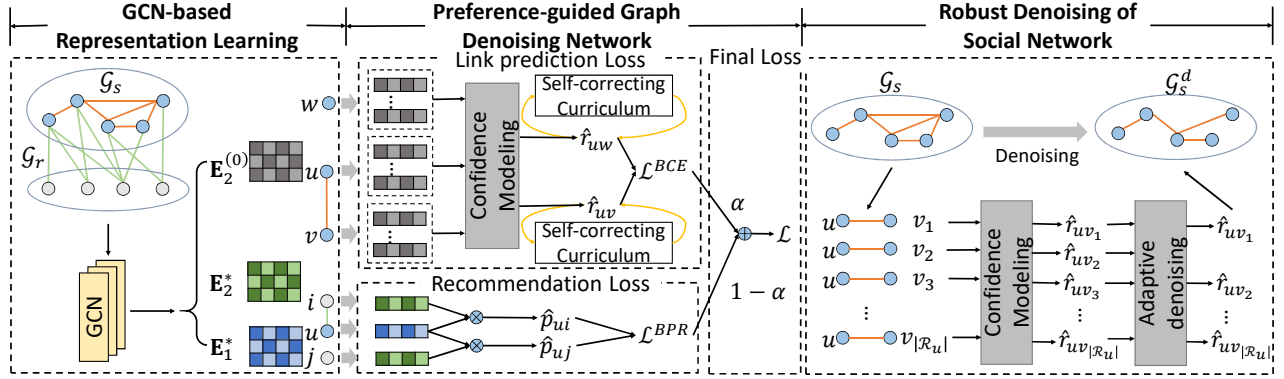


Figure 2: Details of graph denoising process in GDMSR.

training the model, making it infeasible to identify informative links from massive data. Current social recommendation models expect to automatically capture the diverse social influence by using a learnable message-passing mechanism (like GAT), which can be ineffective because of indirect supervision from recommendation loss. Since the purpose of denoising is to obtain a sparsified but more informative social graph for user preference learning, we propose to design a preference-guided graph denoising network for solving the social relation redundancy issue in the context of social recommendation.

**GCN-based representation learning.** Similar to GNN based social recommendation models, our designed preference-guided graph denoising network in GDMSR also adopts a GNN based model structure, with the same input of both social graph  $\mathcal{G}_s$  and interaction graph  $\mathcal{G}_r$ . Specifically, the user and item representations after  $K$ -hop propagation, i.e.,  $\{\mathbf{E}_1^{(k)}\}_{k=0}^K$  and  $\{\mathbf{E}_2^{(k)}\}_{k=0}^K$ , are calculated as in Equation (1) to (4). Since we aim to improve the previous learning approach via attention mechanism, the specific GNN module is GCN instead of GAT.

**Link prediction training.** Given a prediction of the confidence score between two users  $u$  and  $v$ , i.e.,  $\hat{r}_{uv}$ , the link prediction problem aims to minimize the following binary cross-entropy (BCE) alike loss function:

$$\mathcal{L}^{BCE} = -\sum_{(u,v) \in \mathcal{R}} \log(\sigma(\hat{r}_{uv})) - \sum_{(u,w) \notin \mathcal{R}} \log(1 - \sigma(\hat{r}_{uw})). \quad (7)$$

In terms of leveraging user preference as the guiding signal for social graph denoising, GDMSR has the following two designs to better exploit the above useful knowledge.

**Preference-based relation confidence modeling.** A common choice for calculating relation confidence  $\hat{r}_{uv}$  is to design a scoring function  $\phi(\cdot)$ , i.e.,

$$\hat{r}_{uv} = \phi\left(\{\mathbf{E}_1^{(k)}(u)\}_{k=0}^K, \{\mathbf{E}_1^{(k)}(v)\}_{k=0}^K\right), \quad (8)$$

where  $\phi(\cdot)$  can either be a learnable NN (like MLP) or a non-parametric operation (like inner product). However, we argue that this way of relation confidence modeling is not precise, as the inputs  $\{\mathbf{E}_1^{(k)}\}_{k=0}^K$  may take in unrelated information. To find influential friends in the context of social recommendation, it is intuitive to directly compare their interaction history because these characterize the similarity of their preferences. However, above GCN based user

representations  $\{\mathbf{E}_1^{(k)}\}_{k=0}^K$  embeds a more comprehensive view of user traits, including user profile (i.e.,  $\mathbf{E}_1^{(0)}$ ), a fusion of user preference and social influence (i.e.,  $\mathbf{E}_1^{(1)}$ ) and higher order ones. To ensure simple but effective guidance from user preferences, our proposed GDMSR only use the interaction history to characterize the relation confidence between two friends. Mathematically,  $\hat{r}_{uv}$  is calculated as follows,

$$\hat{r}_{uv} = \text{Trf}\left(S_L\left(\{\mathbf{E}_2^{(0)}(i) | \forall i \in \mathcal{P}_u\}\right) \oplus S_L\left(\{\mathbf{E}_2^{(0)}(j) | \forall j \in \mathcal{P}_v\}\right)\right), \quad (9)$$

where a Transformer module [43], denoted as Trf, is used given its power in modeling similarity between two sequences of user interaction history. Specifically, the input is the concatenation of two sequence embeddings (denoted as  $S_L(\cdot)$ ) representing the interaction history of  $u$  and  $v$ , respectively, each with a fixed length of  $L$ . Since no positional information is required in modeling relation confidence, here we do not apply positional encoding in the Transformer. As for the output, we add a “CLS” alike token at the end of the input sequence and use the corresponding transformer-encoded embedding vector to calculate  $\hat{r}_{uv}$  with a MLP.

**Co-optimization with recommendation loss.** Besides designing the above preference-based relation confidence modeling structure, we further enhance preference guidance by co-optimizing the denoising model with a recommendation loss. As shown in Equation (9), the quality of item embeddings is vital in characterizing relation confidence between users. Motivated by the idea of improving representation quality with self-supervised learning, we propose to add a recommendation loss ( $\mathcal{L}^{BPR}$  in Equation (6)) and simultaneously train item embeddings to predict user preferences, which benefits their use in graph denoising in return. Therefore, the final loss function is as follows,

$$\mathcal{L} = \alpha \mathcal{L}^{BCE} + (1 - \alpha) \mathcal{L}^{BPR}, \quad (10)$$

where  $\alpha \in [0, 1]$  is a hyperparameter that controls the relative importance of link prediction loss and recommendation loss.

In a word, our proposed GDMSR can denoise social relations by exploiting user preference in a two-fold manner, i.e., explicit preference-based relation modeling and implicit co-supervision from preference learning, which both help solve the social relation redundancy issue.

### 3.3 Robust Denoising of Social Network

Another key challenge of denoising social network for recommendation is the existence of noisy social relations (i.e., friends with no shared preference.) that increases the learning difficulty of the denoising model. For example, suppose there are two connected users  $\{u, v\}$  with distinct preferences, having two sets of interacted items  $\{\mathcal{P}_u, \mathcal{P}_v\}$  that are less correlated with each other, pushing  $\hat{r}_{uv}$  in Equation (9) to a large value is generally difficult and may induce a biased model that has to memorize this difficult sample [1]. Therefore, the proposed GDMSR designs a self-correcting curriculum learning mechanism and an adaptive denoising strategy to alleviate this noisy effect and thus achieve robust graph denoising.

**Self-correcting curriculum learning mechanism.** during the GDMSR learning process, relations with a high probability of being noisy, i.e., with a rather low confidence score predicted by the current denoising model, are removed, and the set of removed relations maintains a dynamic update with the model every several training epochs. This equals following a difficulty-based curriculum that tends to favor the rather easier samples so as to achieve robust learning under noise and keep correcting this curriculum by replacing the more difficult samples with the easier ones according to the current model [64]. Specifically, for a user  $u$ , the last  $\eta_u$  of her friends ordered by  $\hat{r}_{uv}$  are considered as noisy relations to remove from training. This set is updated every  $D$  epochs, where  $D$  is a hyperparameter representing the length of each period in the above self-correcting curriculum.

**Adaptive denoising strategy.** As illustrated in Figure 2, the trained denoising model is used in the final stage for social network denoising, i.e., predicting the confidence score of each existed social relation and removing those with low confidence. On the one hand, since the training process with noisy labels may be unstable, the robustness of GDMSR is further enhanced by smoothing its prediction results along the training process [5, 22]. Specifically,  $\hat{r}_{uv}$  at the end of  $k$ th curriculum period, i.e.,  $kD$ th epoch, is smoothed by that of the last period. Mathematically,  $\hat{r}_{uv}$  is updated as follows,

$$\hat{r}_{uv}(t = kD) = \beta \cdot \hat{r}_{uv}(t = (k - 1)D) + (1 - \beta) \cdot \hat{r}_{uv}(t = kD), \quad (11)$$

where a hyperparameter  $\beta$  controls smoothness. On the other hand, Motivated by *Dunbar's number theory* suggesting an upper limit of a user's close friends [12], we propose to adaptively denoise the user  $u$ 's social graph based on her friend number, i.e.,  $|\mathcal{R}_u|$ . Specifically, for each user  $u$ ,  $\eta_u$  of  $u$ 's friends are removed from the original graph  $\mathcal{G}_s$ , where we use the same hyperparameter  $\eta_u$  as in the above curriculum design. Mathematically, the denoising ratio  $\eta_u$  is calculated as follows,

$$\eta_u = \begin{cases} 0, & \text{if } |\mathcal{R}_u| < \epsilon, \\ \lceil [\log_{10}(|\mathcal{R}_u|)]^\gamma \times R \rceil, & \text{else,} \end{cases} \quad (12)$$

where  $\epsilon$ ,  $\gamma$  and  $R$  are three hyperparameters. The core idea behind this formula is that sparsely-connected users ( $|\mathcal{R}_u| < \epsilon$ ) can retain all their relations while densely-connected users get to cut more relations in the denoised graph  $\mathcal{G}_s^d$ , which is more robust comparing uniform dropping regardless of friend number.

In a word, by following the above self-correcting curriculum during the training process and adopting an adaptive manner of denoising, our proposed GDMSR is able to robustly denoise social relations based on user preferences. So far, we have completed the

entire denoising process, and the whole algorithm is shown in the Appendix A.

## 4 EXPERIMENTS

### 4.1 Experiment Settings

**Dataset.** We conduct experiments on three open social recommendation datasets, including Ciao, Yelp and Douban. These datasets have been widely used in social recommendation related works [50, 60]. More dataset preprocessing details are in the Appendix B. The statistical information of each dataset is shown in Table 1.

**Baseline.** Since our GDMSR is able to generate denoised social networks and adapt to arbitrary social recommendation models. For social recommendation model, we choose Diffnet++ [50] and MHCN [60] as baselines. For the denoising method, we compare GDMSR with Rule based approach, NeuralSparse [63] and ESRF [59]. Finally, we also add LightGCN [20] as a baseline, which only uses interaction data without social relation data. More about the baseline method is in the Appendix B.

**Evaluation.** To evaluate the performance of all methods, we use  $Recall@K$  and  $NDCG@K$  as metrics, where  $K=\{1,3\}$ . According to the suggestions of [62], we use *real-plus-N* [3, 35] to calculate the measures. For each user in the test set, we randomly sample 100 items that the user has not interacted with and rank them with the positive samples in the test set.

More implementation details are in the Appendix B.

### 4.2 Overall Performance

We compare the performance of GDMSR with original social recommendation models without social network denoising and two denoising methods. The results are shown in Table 2. The overall average denoising ratio for all denoising methods ranges from about 5% to 10% depending on the hyperparameters. All results are average value after five repeat experiments. We performed paired t-test between the results of GDMSR and best-performed baseline. All improvements are statistically significant for  $p < 0.01$ . It can be observed that GDMSR achieves the best performance in all experiments and the max improvement is up to 10% on one dataset, i.e., Ciao.

Further, we have the following findings. **First**, compared to using the original social network directly, after denoising the social network through different methods, the performance is improved in all experiments. This shows that in social recommendation, unreliable social relations in social networks are ubiquitous, and not all social relations reflect the homogeneity or social influence of users. Even a simple ruled based denoising method can improve the performance. **Second**, compared with NeuralSparse which uses both user interaction information and social information for denoising, GDMSR has better performance. This shows that in complicated social networks, it is more effective to use only user preference than to use social relations and user preference for denoising. **Finally**, the proposed GDMSR can achieve performance improvements on the basis of different models. For example, on the Douban dataset, GDMSR achieved performance improvement of more than 2% compared to the best baseline, using both Diffnet++ and MHCN, which means that GDMSR can identify useless relations without relying

**Table 1: Basic information of datasets.**

Dataset	#Users	#Items	#Interactions	#Relations	Interaction Density	Relation Density
Ciao	7,355	17,867	140,628	111,679	0.11%	0.21%
Yelp	32,827	59,972	598,121	964,510	0.03%	0.09%
Douban	2,669	15,940	535,210	32,705	1.26%	0.46%

**Table 2: Overall performance of our proposed method on different recommendation methods.**

Dataset		Ciao			Yelp			Douban		
Basemodel	Method	R@1	R@3	N@3	R@1	R@3	N@3	R@1	R@3	N@3
LightGCN	-	0.2298	0.0785	0.2071	0.5861	0.2774	0.5804	0.4321	0.1696	0.4156
Diffnet++	w/o denoising	0.2742	0.1109	0.2639	0.6031	0.3072	0.5897	0.5165	0.2156	0.4988
	Rule based	0.2860	0.1123	0.2677	0.6230	0.3228	0.5996	0.5358	<u>0.2489</u>	0.5172
	NeuralSparse	<u>0.2869</u>	0.1153	0.2734	<u>0.6383</u>	<u>0.3289</u>	<u>0.6054</u>	<u>0.5470</u>	0.2226	0.5102
	ESRF	0.2864	<u>0.1197</u>	<u>0.2736</u>	0.6184	0.3124	0.5958	0.5374	0.2393	<u>0.5194</u>
	GDMSR	<b>0.3020</b>	<b>0.1244</b>	<b>0.2821</b>	<b>0.6449</b>	<b>0.3291</b>	<b>0.6102</b>	<b>0.5614</b>	<b>0.2540</b>	<b>0.5297</b>
	$\Delta$	5.26%	3.93%	3.11%	1.03%	0.06%	0.79%	2.63%	2.05%	1.98%
MHCN	w/o denoising	0.2330	0.0884	0.2297	0.6991	0.3252	<u>0.6364</u>	0.6198	0.3167	0.5933
	Rule based	0.2301	0.0916	0.2311	0.6966	0.3234	0.6347	0.6082	<u>0.3372</u>	0.5980
	NeuralSparse	0.2461	<u>0.1034</u>	0.2540	<u>0.7012</u>	0.3288	0.6352	<u>0.6206</u>	0.3349	<u>0.6011</u>
	ESRF	<u>0.2495</u>	0.1028	<u>0.2568</u>	0.6927	<u>0.3298</u>	0.6344	0.6194	0.3244	0.5995
	GDMSR	<b>0.2618</b>	<b>0.1138</b>	<b>0.2632</b>	<b>0.7036</b>	<b>0.3405</b>	<b>0.6434</b>	<b>0.6396</b>	<b>0.3496</b>	<b>0.6137</b>
	$\Delta$	4.93%	10.06%	2.50%	0.34%	3.24%	1.10%	3.06%	3.68%	2.10%

on specific models and can be more widely adapted to different social recommendation models.

### 4.3 Ablation Study

**Preference-guidance from co-optimization.** We propose to jointly optimize the recommendation loss and link prediction loss during denoising training, thereby improving the accuracy of relation confidence modeling. We tested the effect of co-optimization under different weights( $\alpha$ ), and the results are shown in Figure 3(a)(b). It can be found that, on different social recommendation models, as  $\alpha$  increases from 0 to 1, the performance first increases and then decreases. In detail, on the Ciao dataset, When  $\alpha = 0.5$ , the three social recommendation models have achieved their respective best performance. On the Douban dataset,  $\alpha = 0.3$  or  $0.5$  works best. This proves that our co-optimization module can obtain better item representation through recommendation loss and further improve the accuracy of the link prediction task. At the same time, when  $\alpha = 0$ , the performance of the model has no obvious advantage compared with the method w/o denoising in Table 2, which shows that it is not enough to only model user preferences, and the denoising model needs the guidance of the link prediction task.

**Modeling relation confidence with only preferences.** In the relation modeling task, we propose to measure the confidence of the relation between users by only using the user’s feedback, that is, the item representations that the user has interacted with. To verify the effect of this idea, we compare our proposed transformer-based relation confidence modeling module with other three options using different information, including using user representations or pooling the item representation directly. The results are shown in

Figure 3(c)-(d). “User-0” and “User-1” indicate using the representation of user nodes in GCN before convolution and from the first layer respectively. We can observe that only using item representation works best, demonstrating the effectiveness of our proposed relation modeling structure, as it ensures simple but effective guidance from user preference (interaction history). At the same time, compared to using user representation, only using item representation for pooling can achieve similar results. This again shows that the denoising model cannot effectively model the users’ social relation confidence when integrating the user-side information including their profile (*i.e.*,  $\mathbf{E}_1^{(0)}$ ) and social influence (*i.e.*,  $\mathbf{E}_1^{(1)}$ ).

**Denoising robustness.** In robust denoising module, we propose an adaptive denoising strategy, which makes the denoising ratio of users with more friends larger. To verify the effect of this strategy, we compare strategies that use the same denoising ratio ( $\eta_u$ ) for all users. As shown in Figure 4 and Appendix B.8, the performance *w.r.t.* *Recall@1* in different cases consistently shows a trend of first increasing and then decreasing, indicating that proper denoising of social network can improve effectiveness of social influence, thereby improving the recommendation performance. Meanwhile, compared with non-adaptive denoising, the proposed GDMSR significantly outperforms in nearly all cases with different denoising ratio. More importantly, our proposed GDMSR is able to achieve further sparsification without hurting original recommendation accuracy, which produce a memory-efficient social graph that is vital in practical applications. Specifically, GDMSR combined with Diffnet++ is able to achieve 20-40% of relation reduction while maintaining an accuracy improvement. Comparatively, this reduction value drops to about 10% when combined with MHCN, which is still of large practical value in commercial recommender

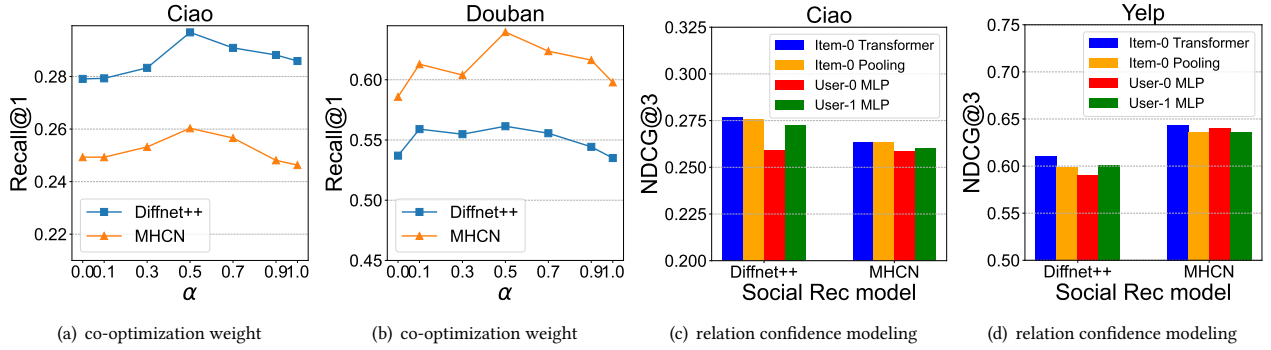


Figure 3: (a)-(b) Performance on different  $\alpha$  for co-optimization. (c)-(d) Performance comparison among different relation confidence modeling structures.

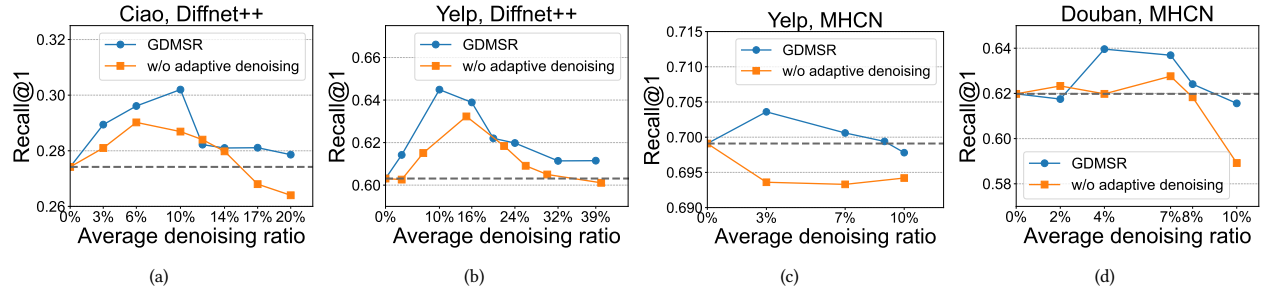


Figure 4: Analysis of denoising robustness with respect to recommendation accuracy under different denoising ratio.

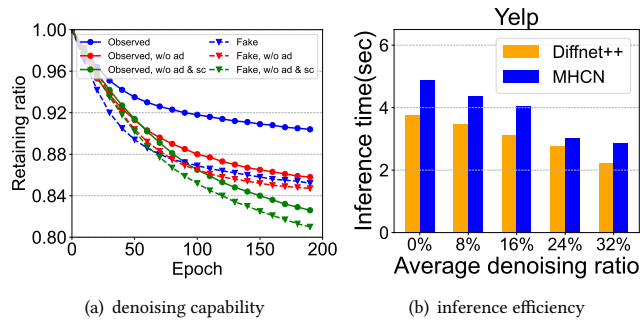


Figure 5: (a) Retaining ratio of social relations adopting different denoising strategies (synthetic). (b) Efficiency comparison w.r.t. social graph related inference time (Yelp).

systems. Since MHCN relies on mining social motifs and leverages hypergraph convolution model to characterize their effects, graph denoising may bring larger impacts on MHCN because of structural changes. In addition, we also compare the effect of choosing different value of hyperparameter  $\gamma$ . Due to space limit, please refer to Appendix B.8 for details.

#### 4.4 Denoising Capability on Synthetic Data

To further evaluate denoising capability of our proposed GDMSR, we construct a synthetic dataset (based on Ciao) with both observed social relations and fake social relations (*i.e.*, we randomly set some unconnected user pairs to be connected) and evaluate

whether GDMSR is able to learn a powerful graph denoising model that can effectively discriminate a given relation label. Specifically, we inject the fake social relations with the same number of those observed ones, and further combine these corrupted relation data with original interaction data to train GDMSR. Then we record the retaining ratio of each part of relation data during the denoising process, and present the results in Figure 5(a). Here we also compare with two variants, *i.e.*, without adaptive denoising (shorten as “w/o ad”) and without adaptive denoising and self-correcting curriculum learning (shorten as “w/o ad & sc”). The results clearly indicate that GDMSR, even though under explicit noise of corrupted relation labels, can still maintain learning robustness by removing more fake relations (16%) than observed relations (9%). In other words, since we inject rather strong noise (50% of relation labels are corrupted), GDMSR manages to achieve a moderately good accuracy of 64% (16/(16+9)), while other two options cannot produce a workable denoising module under this level of label noise.

#### 4.5 Practicality Evaluation

**Computation efficiency.** Our proposed GDMSR can produce a much sparser social graph (*e.g.*, 20-40% smaller on Yelp), which can improve computation efficiency in inference period. In Figure 5(b) we present an inference time analysis of social graph related computations (social-network-side GNN), which is conducted on Yelp dataset by inferring 50w records. The results clearly demonstrate the increasing improvement of computation efficiency as denoising ratio goes larger.

**Table 3: Overall performance of GDMSR based on 30% of interaction data.**

Dataset		Ciao			Yelp			Douban		
Basemodel	Method	R@1	R@3	N@3	R@1	R@3	N@3	R@1	R@3	N@3
Diffnet++	w/o denoising	0.2742	0.1109	0.2639	0.6031	0.3072	0.5897	0.5165	0.2156	0.4988
	GDMSR	<b>0.2899</b>	<b>0.1175</b>	<b>0.2755</b>	<b>0.6305</b>	<b>0.3189</b>	<b>0.6021</b>	<b>0.5490</b>	<b>0.2497</b>	<b>0.5295</b>
	$\Delta$	5.73%	5.95%	4.40%	4.54%	3.81%	2.10%	6.29%	15.82%	6.15%
MHCN	w/o denoising	0.2330	0.0884	0.2297	0.6991	0.3252	0.6364	<b>0.6198</b>	0.3167	0.5933
	GDMSR	<b>0.2616</b>	<b>0.1093</b>	<b>0.2638</b>	<b>0.7005</b>	<b>0.3438</b>	<b>0.6445</b>	0.6148	<b>0.3360</b>	<b>0.6012</b>
	$\Delta$	12.27%	23.64%	14.85%	0.20%	5.72%	1.27%	-0.81%	6.09%	1.33%

**Zero-shot denoising without frequent training.** In industry practice, it is rather time consuming and nearly infeasible to frequently update graph denoising model given the massive social relation data (up to hundreds of millions in Facebook and Wechat). Therefore, GDMSR is expected to support denoising training in a less frequent manner, *i.e.*, produce high-quality distilled social graph without training on newly coming interaction data [28]. To evaluate this practicality, we design another experiment by only using 30% of interaction data to train denoising module of GDMSR and then use all interaction data as input to denoise social graph. This setting simulates the fact that GDMSR cannot be timely updated in practice. Instead, we may use the old GDMSR model but the new interaction data to directly denoise current social graph in a zero-shot-like manner. Then, the denoised graph is used to train the social recommendation models and results are listed in Table 3. It can be observed that GDMSR is still able to effectively denoise the social network even when only 30% of the training samples are used. In most cases, the recommendation accuracy can be improved compared to without denoising, with the highest improvement reaching 23%, demonstrating that GDMSR has high scalability in terms of industrial practice. In fact, this is mainly owing to our design of preference-guide denoising with only interactions, which is more robust and less influenced by other contextual changes.

## 5 RELATED WORK

### 5.1 Graph Denoising for Social Recommendation

Early works use simple statistics like number of co-interactions [31] to indicate the confidence degree of social relations, which is generally ineffective due to data sparsity [40]. In order to capture observed diversity of social influence, attention mechanism [50, 55], fine-grained contextual information [14], expectation-maximization method [47] or reinforcement learning [11] can be leveraged to learn adaptive weights among various friends. Although above techniques can be applied in prevalent GSocRec models, they generally fall short of learning the degree of social influence effectively due to a lack of groundtruth labels. From a view of general graph machine learning, graph sparsification methods [7, 63] learn to drop edges from a general graph so as to achieve robust learning, which does not consider the specific social recommendation context. Moreover, graph structure learning methods [29, 30, 61] are optimized to not only drop but also add edges, which are too time-consuming for social recommendation as they generally operate on a huge space with cartesian product between users. As a specific extension in

social context, there are a series of works that propose to not only remove observed social relations with low quality but also add a few potential relations [24, 56, 57, 59]. This idea has also been adopted in dynamic recommendation scenarios like session recommendation by starting with observed social relations and updating relations during training process [38, 49]. In addition, the effectiveness of social recommendation can also be improved by using contextual features [26] or multi-view learning [25]. Different from above works, our proposed GDMSR method can produce a distilled social graph, not only sparser but also more informative, that is more suitable for real-world social recommender systems. In experiments we also compare with one competitive method above [59] to demonstrate effectiveness of GDMSR.

### 5.2 Denoising for Recommendation

In general recommender systems, another trending topic is feedback denoising that enables more robust preference learning [21, 42]. Since there exists heterogeneity in users' manner of expressing preferences, it is non-optimal to set a unitary threshold for either explicit feedback or implicit feedback. For example, users differ in their criterion of rating items with low scores [45] or not clicking [8–10]. There are also some works to solve this problem through causal inference [16]. To deal with this problem of learning from noisy labels [37], existing works adopt techniques like adding graph-based priors [39], cross-model agreements [48] and graph sparsification [17] to denoise observed user-item interaction data. Different from above works, we focus on denoising social network, *i.e.*, user-user relation data, so as to leverage power of social homophily and social influence in a more effective and efficient way.

## 6 CONCLUSION AND FUTURE WORK

In this paper, we approach the problem of graph based social recommendation by rethinking the reliability of the social network for learning GNN recommendation models. Since there is no reliably labeled data, we choose to utilize user-item feedback that reflects user preference to filter noisy social relations, which, in turn, improves the preference learning. Experiments on three public benchmark datasets and different social recommendation models not only show the effectiveness of the denoising method but also demonstrate our proposed solution can serve as a general framework. Precisely, the improvement is steady and significant, up to 10%.

As for the future plan, one interesting direction is to deeply analyze the difference between the original social network and the denoised one, supported by the social network theory [4].



## ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under 2020AAA0106000, and the National Natural Science Foundation of China under 61971267, U1936217, 61972223, 62272262.

## REFERENCES

- [1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *ICML*. 233–242.
- [2] Eytan Bakshy, Dean Eckles, Rong Yan, and Itamar Rosenn. 2012. Social influence in social advertising: evidence from field experiments. In *ACM conference on electronic commerce*. 146–161.
- [3] Alejandro Bellogin, Pablo Castells, and Ivan Cantador. 2011. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Recsys*. 333–336.
- [4] Damon Centola. 2010. The spread of behavior in an online social network experiment. *science* 329, 5996 (2010), 1194–1197.
- [5] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. 2021. Beyond Class-Conditional Assumption: A Primary Attempt to Combat Instance-Dependent Label Noise. In *AAAI*, Vol. 35. 11442–11450.
- [6] Renata Gonçalves Curty and Ping Zhang. 2011. Social commerce: Looking back and forward. *Proceedings of the American Society for Information Science and Technology* 48, 1 (2011), 1–10.
- [7] Anyan Dai, Wei Jin, Hui Liu, and Suhang Wang. 2022. Towards robust graph neural networks for noisy graphs with sparse labels. In *WSDM*. 181–191.
- [8] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data.. In *IJCAI*. Macao, 2230–2236.
- [9] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *NeurIPS* 33 (2020), 1094–1105.
- [10] Jingtao Ding, Guanghui Yu, Xiangnan He, Fuli Feng, Yong Li, and Depeng Jin. 2019. Sampler design for bayesian personalized ranking by leveraging view data. *IEEE transactions on knowledge and data engineering* 33, 2 (2019), 667–681.
- [11] Qihan Du, Li Yu, Huiyuan Li, Youfang Leng, Ningrui Ou, and Junyao Xiang. 2022. Denoising-Guided Deep Reinforcement Learning For Social Recommendation. In *ICASSP*. 4113–4117.
- [12] Robin Dunbar. 2010. *How many friends does one person need?: Dunbar's number and other evolutionary quirks*. Faber & Faber.
- [13] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*. 417–426.
- [14] Bairan Fu, Wenming Zhang, Guangneng Hu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2021. Dual side deep context-aware modulation for social recommendation. In *WWW*. 2524–2534.
- [15] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2022. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Transactions on Recommender Systems* (2022).
- [16] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2022. Causal Inference in Recommender Systems: A Survey and Future Directions. *arXiv preprint arXiv:2208.12397* (2022).
- [17] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng. 2022. Self-Guided Learning to Denoise for Robust Recommendation. *arXiv preprint arXiv:2204.06832* (2022).
- [18] 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 *AAAI*, Vol. 29.
- [19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *NIPS* 30 (2017).
- [20] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.
- [21] Yue He, Yancheng Dong, Peng Cui, Yuhang Jiao, Xiaowei Wang, Ji Liu, and Philip S Yu. 2021. Purify and Generate: Learning Faithful Item-to-Item Graph from Noisy User-Item Interaction Behaviors. In *KDD*. 3002–3010.
- [22] Lang Huang, Chao Zhang, and Hongyang Zhang. 2020. Self-adaptive training: beyond empirical risk minimization. *NeurIPS* 33 (2020), 19365–19376.
- [23] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [24] Adit Krishnan, Hari Cheruvu, Cheng Tao, and Hari Sundaram. 2019. A modular adversarial approach to social recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1753–1762.
- [25] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Fei Wang, Hao Yang, and Hari Sundaram. 2022. Multi-task Knowledge Graph Representations via Residual Functions. In *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part I*. Springer, 262–275.
- [26] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. 2020. Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1081–1090.
- [27] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58, 7 (2007), 1019–1031.
- [28] Dandan Lin, Shijie Sun, Jingtao Ding, Xuehan Ke, Hao Gu, Xing Huang, Chong-gang Song, Xuri Zhang, Lingling Yi, Jie Wen, et al. 2022. PlatoGL: Effective and Scalable Deep Graph Learning System for Graph-enhanced Real-Time Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3302–3311.
- [29] Nian Liu, Xiao Wang, Lingfei Wu, Yu Chen, Xiaojie Guo, and Chuan Shi. 2022. Compact Graph Structure Learning via Mutual Information Compression. In *WWW*. 1601–1610.
- [30] Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. 2022. Towards Unsupervised Deep Graph Structure Learning. *arXiv preprint arXiv:2201.06367* (2022).
- [31] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*. 287–296.
- [32] Peter V Marsden and Noah E Friedkin. 1993. Network studies of social influence. *Sociological Methods & Research* 22, 1 (1993), 127–151.
- [33] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.
- [34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.
- [35] Alan Said and Alejandro Bellogin. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Recsys*. 129–136.
- [36] Chijun Sima, Yao Fu, Man-Kit Sit, Liyi Guo, Xuri Gong, Feng Lin, Junyu Wu, Yongsheng Li, Haidong Rong, Pierre-Louis Aublin, et al. 2022. Ekko: A {Large-Scale} Deep Learning Recommender System with {Low-Latency} Model Update. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*. 821–839.
- [37] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE TNNS*.
- [38] Liqiang Song, Ye Bi, Mengqiu Yao, Zhenyu Wu, Jianming Wang, and Jing Xiao. 2020. Dream: A dynamic relation-aware model for social recommendation. In *CIKM*. 2225–2228.
- [39] Jonathan Strahl, Jaakko Peltonen, Hirsohi Mamitsuka, and Samuel Kaski. 2020. Scalable probabilistic matrix factorization with graph-based priors. In *AAAI*, Vol. 34. 5851–5858.
- [40] Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. *Social Network Analysis and Mining* 3, 4 (2013), 1113–1133.
- [41] Ye Tao, Ying Li, Su Zhang, Zhirong Hou, and Zhonghai Wu. 2022. Revisiting Graph based Social Recommendation: A Distillation Enhanced Social Graph Network. In *WWW*. 2830–2838.
- [42] Changxin Tian, Yuexiang Xie, Yaliang Li, Nan Yang, and Wayne Xin Zhao. 2022. Learning to Denoise Unreliable Interactions for Graph Collaborative Filtering. In *SIGIR*. 122–132.
- [43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NIPS* 30 (2017).
- [44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [45] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *WSDM*. 373–381.
- [46] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social recommendation with strong and weak ties. In *CIKM*. 5–14.
- [47] Xin Wang, Wenwu Zhu, and Chenghao Liu. 2019. Social recommendation with optimal limited attention. In *KDD*. 1518–1527.
- [48] Yu Wang, Xin Xin, Zaiqiao Meng, Joemon M Jose, Fuli Feng, and Xiangnan He. 2022. Learning Robust Recommenders through Cross-Model Agreement. In *WWW*. 2015–2025.
- [49] Chunyu Wei, Bing Bai, Kun Bai, and Fei Wang. 2022. GSL4Rec: Session-based Recommendations with Collective Graph Structure Learning and Next Interaction Prediction. In *WWW*. 2120–2130.
- [50] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [51] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *SIGIR*. 235–244.

- [52] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. 2018. Socialgen: An efficient graph convolutional network based model for social recommendation. *arXiv preprint arXiv:1811.02815* (2018).
- [53] 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 *WWW*. 2091–2102.
- [54] Fengli Xu, Zhenyu Han, Jinghua Piao, and Yong Li. 2019. "I Think You'll Like It" Modelling the Online Purchase Behavior in Social E-commerce. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.
- [55] Liangwei Yang, Zhiwei Liu, Yingdong Dou, Jing Ma, and Philip S Yu. 2021. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *SIGIR*. 2141–2145.
- [56] Junliang Yu, Min Gao, Jundong Li, Hongzhi Yin, and Huan Liu. 2018. Adaptive implicit friends identification over heterogeneous network for social recommendation. In *CIKM*. 357–366.
- [57] Junliang Yu, Min Gao, Hongzhi Yin, Jundong Li, Chongming Gao, and Qinyong Wang. 2019. Generating reliable friends via adversarial training to improve social recommendation. In *ICDM*. 768–777.
- [58] 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 *KDD*. 2084–2092.
- [59] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2020. Enhance social recommendation with adversarial graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [60] 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 *WWW*. 413–424.
- [61] Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. 2021. Heterogeneous graph structure learning for graph neural networks. In *AAAI*.
- [62] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *CIKM*. 2329–2332.
- [63] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. 2020. Robust graph representation learning via neural sparsification. In *ICML*. PMLR, 11458–11468.
- [64] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. 2020. Error-bounded correction of noisy labels. In *ICML*. PMLR, 11447–11457.

## A TRAINING PROCESS OF GDMSR

The overall process of GDMSR is shown in Algorithm 1.

---

### Algorithm 1: Training process of GDMSR

---

**Require:** social relations set  $\mathcal{R}$ , interactions set  $\mathcal{P}$

- 1: **for**  $epoch = 1$  to  $MAX\_EPOCH$  **do**
- 2:   Generate training batch  $\mathcal{B}$  from  $\mathcal{P}$
- 3:   **for**  $(u, i)$  in  $\mathcal{B}$  **do**
- 4:     sample  $(u, j), (u, v)$  and  $(u, w)$  with  $j \notin \mathcal{P}_u, v \in \mathcal{R}_u$  and  $w \notin \mathcal{R}_u$
- 5:     Calculate recommendation loss  $\mathcal{L}^{BPR}$  by Eq.6
- 6:     Calculate link prediction loss  $\mathcal{L}^{BCE}$  by Eq.7
- 7:     Calculate final loss  $\mathcal{L}$  and update parameter by Eq.9
- 8:   **end for**
- 9:   **if**  $epoch \% D == 0$  **then**
- 10:     **for**  $(u, v)$  in  $\mathcal{R}$  **do**
- 11:       Calculate  $\hat{r}_{uv}(t = kD)$  by Eq.8 and Eq.10
- 12:     **end for**
- 13:     **for**  $u$  in  $\mathcal{U}$  **do**
- 14:       Calculate  $\eta_u$  by Eq.11
- 15:       Remove last  $\eta_u$  of  $u$ 's friends ordered by  $\hat{r}_{uv}(t = kD)$
- 16:     **end for**
- 17:   **end if**
- 18: **end for**

---

## B EXPERIMENT DETAILS

### B.1 Dataset preprocessing

For all datasets, we only retain the samples with a score of 4 or 5 as positive samples. Besides, we filter users and items according to the number of interactions and the number of users' friends and only retain samples related to users and items with high activity. For the Yelp dataset, we retain the user attributes, and since the original data does not contain user attributes for the other two datasets, we only use the user id as a feature.

### B.2 Baselines

Here is an introduction to the baseline methods we used

**Diffnet++.** This model uses graph attention networks to model diverse social influences among various friends.

**MHCN.** This model uses hypergraph convolution methods to capture different impacts of social motifs on preference learning among friends.

**Rule based approach.** Based on the social homogeneity assumption, the similarity between users can be measured according to the items that the users interact with. Therefore, we remove those relations with a few items that users have both interacted with.

**NeuralSparse.** This is an end-to-end adaptive graph sparsification method that learns a differentiable sampler from empirical data by using Gumbel softmax.

**ESRF.** This end-to-end social recommendation framework leverages Generative Adversarial Nets (GAN) so as to generate more densely-connected social graphs for social recommendation.

### B.3 Running Environment

The experiments are conducted on a single Linux server with AMD Ryzen Threadripper 2990WX@3.0GHz, 128G RAM and 3 NVIDIA GeForce RTX 2080TI-11GB. Our proposed GDMSR is implemented in PyTorch 1.10 and Python 3.7. Social recommendation models and baselines including SocialGCN, Diffnet++, MHCN and NeuralSparse are implemented in Tensorflow 1.14 and Python 3.7.

### B.4 Assets

The code of our proposed GDMSR is included in the supplementary materials. Diffnet++<sup>1</sup>, MHCN<sup>2</sup> and NeuralSparse<sup>3</sup> are implemented based on open source code. It should be noted that NeuralSparse is not an official implementation. The implementation of SocialGCN is based on Diffnet++. We removed the attention module and made some corresponding modifications according to the original paper [52].

### B.5 Implementation Detail.

For all experiments, the embedding size and batch size are set to 8 and 1024 respectively, and they all use the Adam optimizer. All social recommendation models use BPR loss. Hyper-parameters like learning rate and dropout are obtained by grid search. The search range is as follows: the learning rate is {0.005, 0.001, 0.0005, 0.0001}, the dropout is [0, 1], and the weights ( $\alpha$ ) for the co-optimization in GDMSR are {0.1, 0.3, 0.5, 0.7, 0.9}. For adaptive denoising strategy, the threshold( $\epsilon$ ) is set to 5, the search range of  $\gamma$  and  $R$  are {0.5, 1.0, 2.0} and {0.01, 0.015, 0.02, 0.025, 0.03} respectively. For NeuralSparse, the number of neighbors sampled by each node is searched from {30, 50, 100}. For all experiments of GDMSR, we train the denoising model for 200 epochs, at which time  $\mathcal{L}^{BCE}$  was basically stable. For all datasets, the interaction history of each user has been pre-processed to have a fixed length, i.e., padding for those shorter than  $L$  and truncation for excess ones. The truncation follows a descending order based on item popularity in datasets.

### B.6 Loss Curve of Denoising Training

In denoising training, for all datasets, we train for 200 epochs so that  $\mathcal{L}^{BCE}$  is stable, and the loss curves are shown in the Figure 6. We can observe that after 200 epochs of training, the loss function has basically stabilized, indicating that the denoising training has been completed.

### B.7 Experiment details

In the experiments, we control the number of social relations that are finally removed by controlling different hyper-parameters for GDMSR and baselines, respectively. Specifically, for GDMSR, we control the  $\epsilon$ ,  $\gamma$  and  $R$ . For Rule based approach, we control the ratio of friends removed per user. For NeuralSparse, We control the hyperparameter  $K$ , which is the maximum number of neighbors that each node can receive message. For ESRF, We also control the number of neighbors each node uses, in the source code, this value is set to 100, and no hyperparameters are provided to tune. We made modifications directly in the source code.

<sup>1</sup><https://github.com/PeiJieSun/diffnet/tree/master/Diffnet%2B%2B>

<sup>2</sup><https://github.com/Coder-Yu/RecQ>

<sup>3</sup><https://github.com/flyingdoog/PTDNet/tree/main/NeuralSparse>

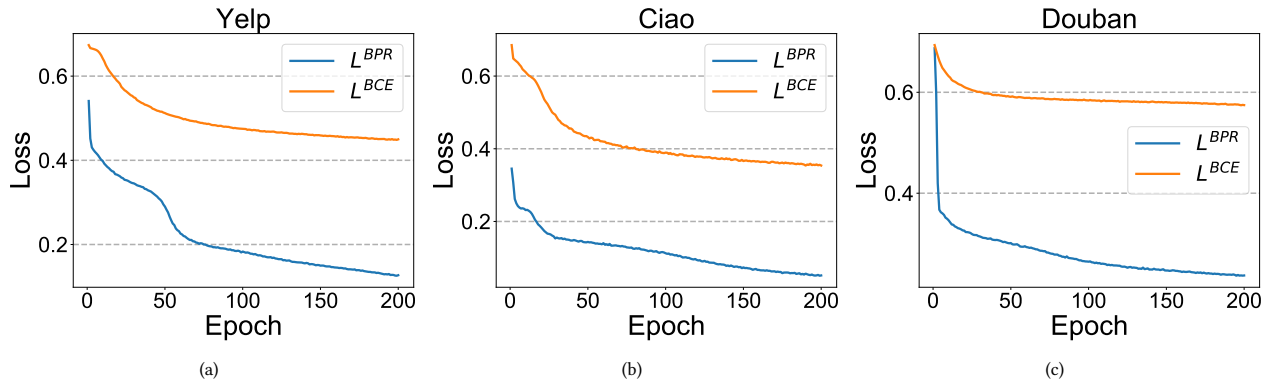


Figure 6: Loss curve of denoising training

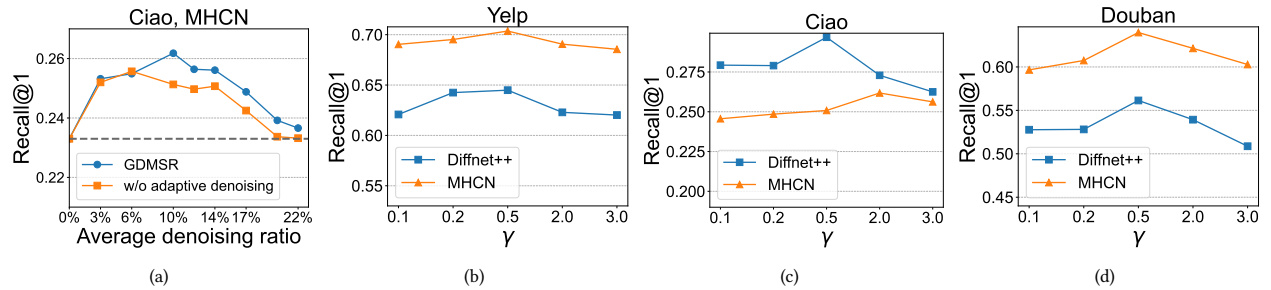


Figure 7: (a)Performance comparison between different denoising strategy. (b)Performance on different  $\gamma$  for adaptive denoising

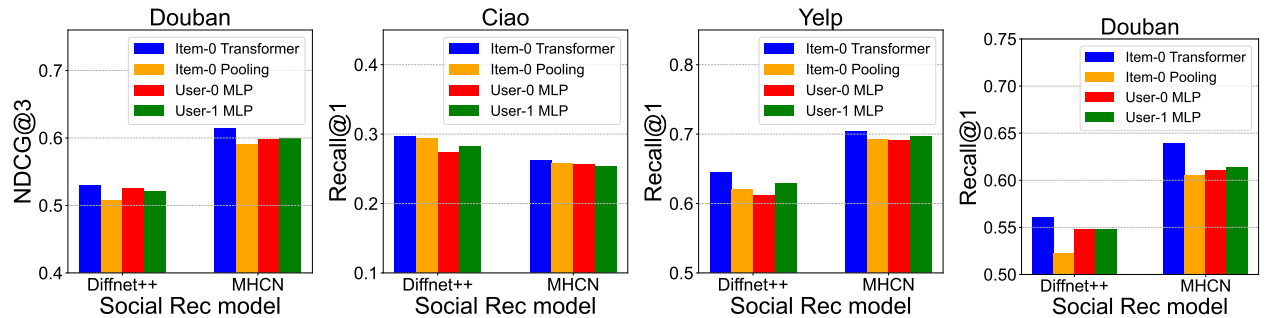


Figure 8: Performance comparison among different relation confidence modeling structures.

### B.8 Denoising Robustness on Yelp

To verify the robustness of the adaptive denoising strategy, we compared the strategy to the method that assign same denoising ratio for all users, and we study the effect of different  $\gamma$  on the

performance. The results are shown in Figure 7(b)-(d). When  $\gamma = 0.5$ , the overall performance is the best. This means that except for those users with very few friends, most users are assigned a larger denoising ratio to achieve better performance.