Dual-Domain Collaborative Denoising for Social Recommendation

Wenjie Chen, Yi Zhang, Honghao Li, Lei Sang, and Yiwen Zhang

***Abstract*—Social recommendation leverages social network to complement user–item interaction data for recommendation task, aiming to mitigate the data sparsity issue in recommender systems. The information propagation mechanism of graph neural networks (GNNs) aligns well with the process of social influence diffusion in social network, thereby can theoretically boost the performance of recommendation. However, existing social recommendation methods encounter the following challenge: both social network and interaction data contain substantial noise, and the propagation of such noise through GNNs not only fails to enhance recommendation performance but may also interfere with the model's normal training. However, despite the importance of denoising for social network and interaction data, only a limited number of studies have considered the denoising for social network and all of them overlook that for interaction data, hindering the denoising effect and recommendation performance. Based on this, we propose a novel model called dual-domain collaborative denoising for social recommendation (DCDSR). DCDSR comprises two primary modules: the structure-level collaborative denoising module and the embedding-space collaborative denoising module. In the structure-level collaborative denoising module, information from the interaction domain is first employed to guide social network denoising. Subsequently, the denoised social network is used to supervise the denoising of interaction data. The embedding-space collaborative denoising module devotes to resisting the noise cross-domain diffusion problem through contrastive learning with dual-domain embedding collaborative perturbation. Additionally, a novel contrastive learning strategy, named Anchor-InfoNCE, is introduced to better harness the denoising capability of contrastive learning. The model is jointly trained under a recommendation task and a contrastive learning task. Evaluating our model on three real-world datasets verifies that DCDSR has a considerable denoising effect, thus outperforms the state-of-the-art social recommendation methods.**

***Index Terms*—Collaborative denoising, graph contrastive learning (GCL), graph neural networks (GNNs), social recommendation.**

Received 30 April 2024; revised 26 December 2024; accepted 6 January 2025. Date of publication 28 January 2025; date of current version 2 October 2025. This work was supported in part by the National Science Foundation of China under Grant 62272001 and Grant 62206002, in part by Hefei Key Common Technology Project under Grant GJ2022GX15, in part by Xunfei Zhiyuan Digital Transformation Innovation Research Special for Universities under Grant 2023ZY001, and in part by Hefei City Project for Selecting the Best Candidates to Undertake the Key Technology Research under Grant 2023SGJ014. *(Corresponding author: Yiwen Zhang.)*

The authors are with the School of Computer Science and Technology, Anhui University, Hefei, Anhui 230601, P.R. China (e-mail: wenjie@stu.ahu.edu.cn; zhangyi.ahu@gamil.com; salmon1802li@gmail.com; sanglei@ahu.edu.cn; zhangyiwen@ahu.edu.cn).

Digital Object Identifier 10.1109/TCSS.2025.3529706

I. Introduction

SOCIAL recommendation [1], [2] has become a significant part of recommender systems due to the rapid development of social platforms, where social network [3], [4] plays an auxiliary role in recommedation task except user-item interaction data. Following the social homophily theory [5] that users with connections in social network tend to interact with similar items, it effectively alleviates the problem of data sparsity [6], [7] in recommender systems.

With the emergence of graph neural networks (GNNs), graph-based social recommendation (GSR) has become a mainstream paradigm [formalized as Fig. 1(a)] for social recommendation. This is attributed to the message propagation mechanism of GNNs, which closely aligns with the social influence process [8], leading to the proposal of a series of GSR models [9], [10], [11], [12], [13], [14], [15], [16]. Following the success of graph contrastive learning (GCL) across various domains, GSR leverages GCL to enable social network to provide supervised signals for interaction data [17], [18], [19]. However, a critical issue has been overlooked in these works, i.e., the presence of noise in social recommendation, encompassing both social network and interaction data.

First, utilizing social network in recommender systems can often introduce amount of noise. Specifically, due to the challenges such as the difficulty of data collection, the social relations in observable social network are implicit, i.e., represented by 0/1 to indicate whether social users trust each other. However, the trust propensity theory [20] suggests that the establishment of social relations does not necessarily imply that social neighbors share similar preferences. For example, diverse students may form social relations because they are in the same class, yet their preferences can be vastly different. The analysis experiments conducted by DESIGN [21] have demonstrated that such social relations account for the vast majority in the social network. Therefore, directly applying the social network to recommendation task is unwise.

Meanwhile, the noise issue is also present in user-item interaction data. Since user preference information is difficult to obtain directly, most user-item interaction data is implicit [22], i.e., in the form of implicit feedback such as clicking and browsing. However, numerous studies have highlighted the substantial discrepancy between implicit feedback and users' actual interests [23], [24], [25], indicating that the available implicit feedback contains a significant amount of noise that is not relevant to the recommendation task. For example, if a

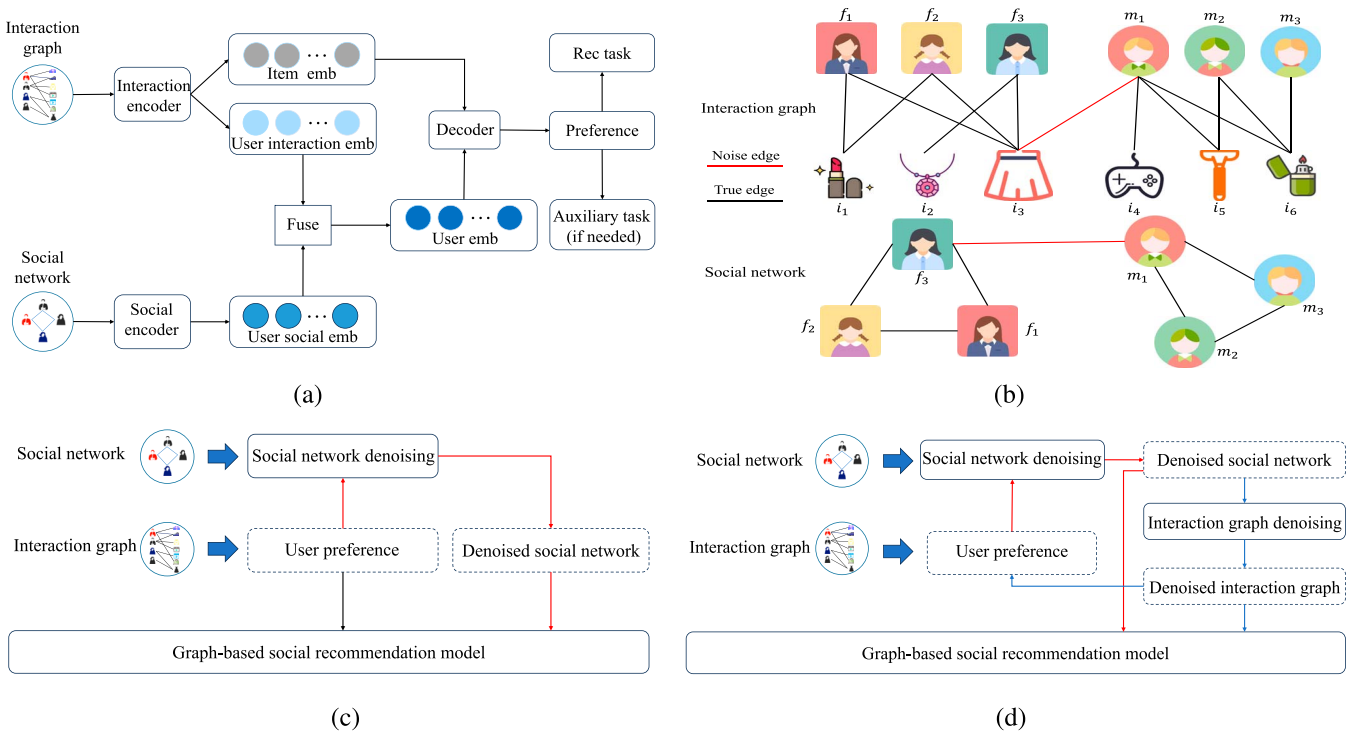

Fig. 1. Mainstream social recommendation paradigms for certain scenario and the motivation example for our work. (a) Graph-based social recommendation paradigm. (b) Motivation example. (c) Preference-guided social network denoising paradigm. (d) Dual-domain collaborative denoising paradigm.

user unintentionally clicks on a certain item page, or watches a certain type of video that he or she would not like due to his/her friend's sharing, this creates noise to implicit feedback.

Despite some efforts such as denoised Self-Augmented Learning paradigm for social recommender systems (DSL) [26] and graph denoising method for social recommendation (GDMSR) [27] have adopted preference-guided approach to denoise social network to improve the robustness of recommendation, they still suffer from a fatal flaw. That is they only focus on the denoising for social network while overlooking the noise issue in interaction data. In fact, within the preference-guided social network denoising paradigm shown in Fig. 1(c), the denoising effect would be vulnerable to the noisy interaction data. As illustrated in Fig. 1(b), there is a three-female user group (f_1, f_2, f_3) and a three-male user group (m_1, m_2, m_3), where the female user group prefers dressing items, and the male group prefers electronic items. However, user m_1 passively clicks information about a dress due to a share from his social neighbor user f_3. This interaction slightly shifts user m_1's preference towards the dressing category, causing his preference to become ambiguous. In this case, the preference-guided social network denoising paradigm would incorrectly ignore the need to intervene with the relation $< f_3, m_1 >$, which impairs the denoising effect of the social network. This issue can be described as noise cross-domain diffusion. Moreover, noisy interaction data would directly lead to suboptimal recommendation performance.

To address the above issues, we propose the model dual-domain collaborative denoising for social recommendation (DCDSR). DCDSR aims to comprehensively address the noise issue to improve the model robustness with a dual-domain collaborative denoising paradigm which is abstracted as Fig. 1(d). Concretely, the structure-level collaborative denoising module utilizes user's preference to perform denoising operations on the original social network, then utilizes the denoised social network in conjunction with knowledge of interaction domain to denoise the interaction graph. The denoised interaction graph can not only promote further denoising for the original social network in a cyclical manner but also positively facilitate the recommendation task. Subsequently, the embedding-space collaborative denoising module performs denoising for user and item embeddings based on GCL with dual-domain embedding collaborative perturbation to allievate the noise cross-domain diffusion issue existing in structure-level denoising process. Furthermore, we propose a novel contrastive learning loss Anchor-InfoNCE (AC-InfoNCE), where the embedding of user/item is set as an anchor, and the positive samples are respectively induced to align with the anchor, while different users/items are induced to converge to uniform distributions in the embedding space. This circumvents the effect of the uncertainty of data augmentation and sample distribution on GCL while expanding the number of positive samples. Above two denoising modules complement each other to cut down the noise present in the original social network and interaction graph, and finally learn high-quality embeddings for recommendation task.

Our contributions can be summarized as follows.

1) We propose a model called dual-domain collaborative denoising for social recommendation (DCDSR). To the best of our knowledge, DCDSR is the first work to address the noise issue present both in social domain and interaction domain with a novel dual-domain collaborative denoising paradigm.
2) We design a structure-level collaborative denoising module which reduces the structural noise by enabling the

TABLE I
NOTATIONS AND EXPLANATIONS

Notation	Explanation
$\mathcal{U}, \mathcal{I}$	The sets of users and items.
M, N	The numbers of users and items.
$\mathcal{G}_R$, $\mathcal{G}_S$	The user-item interaction graph and social network.
$\mathcal{N}_u, \mathcal{S}_u$	The first-order neighbors of user u in $\mathcal{G}_R$ and $\mathcal{G}_S$.
$\mathbf{e}_u, \mathbf{e}_i$	The learnable embedding of user u and item i.
$\mathbf{h}_u^r, \mathbf{h}_i^r$	The preference embedding of user u and item i.
$\mathbf{h}_u^{\text{enhanced}}$	The social-enhanced embedding of user u.
$\mathbf{p}_u^r, \mathbf{p}_u^s$	The interaction embedding and social embedding of user u.
$\mathbf{p}_i^r$	The interaction embedding of item i.
PC, IC	The preference consistency and interaction compatibility.
β_s, β_r	The denoising threshold for social network and interaction graph.
$\lambda_1, \lambda_2, \lambda_3$	The weights of three CL losses.

interaction graph and social network to mutually guide each other's cyclic reconstruction.

3) We design a embedding-space collaborative denoising module to allievate the noise diffusion across interaction and social domains through contrastive learning with dual-domain embedding collaborative perturbation. In addition, a robust contrastive loss function AC-InfoNCE is designed to enhance the contrastive learning for embedding denoising.
4) Extensive experiments conducted on three real-world datasets demonstrate that DCDSR is superior to those state-of-the-art GSR models and denoising GSR models. Furthermore, DCDSR is more noise-resistant when faced with varying degrees of external noise interaction attacks.

II. PRELIMINARY AND RELATED WORKS

In this section, we briefly review graph-based social recommendation (GSR) and graph contrastive learning (GCL), which are the backbones of DCDSR. In addition, for ease of reading and understanding, we list some notations with explanations in Table I.

A. Graph-Based Social Recommendation

GSR is the application of GNNs based on user-item interaction graph and social network to model user and item representations for recommendation tasks. GNNs [28], [29] open the information transfer channel between graph nodes, enabling nodes to aggregate information from their first-order neighbor or higher-order neighbor nodes, thus enriching the features of nodes. In recent years, GNNs have developed rapidly and a series of graph neural network instances have been proposed, such as graph convolutional neural network (GCN) [28], graph attention network (GAT) [30], etc., and utilized in downstream tasks, such as node classification [31], recommender systems [32], traffic prediction [33] and relation extraction [34]. In recommender systems, some GCNs suitable for collaborative filtering task have been proposed, e.g., neural graph collaborative filtering (NGCF) [32], LightGCN [35], and neighbor item embedding-aware graph convolutional network for recommendation (NIE-GCN) [36].

In GSR, user–user social network is incorporated alongside user-item interaction graph to enhance recommendation task, the logic basis of which is that a user's item preferences are simultaneously influenced by their historical item interactions and social relations [1], [2]. Thus the mainstream paradiam of GSR [9], [10], [17], [37], [38], [39] separately models the user interaction embeddings and social embeddings by GNN and then explicitly or implicitly fuses these embeddings to inference user's preference for recommendation task. For example, DiffNet [9] organically fuses the user embeddings with those of its neighboring users via GNN and then sums them with the average embeddings of the item's neighbors to obtain the final user embedding representation. Disentangled modeling of social homophily and influence for social recommendation (DISGCN) [37] decouples the user/item representations by using GNNs to learn the social homophily embeddings of users and the social influence embeddings of items on social network and interaction graph, respectively, and finally fusing these two embeddings. SocialLGN [38] designs a lightweight graph convolutional network-based representation propagation mechanism for both interaction graph and social network. In addition, some studies [40], [41] have also focused on disentangling social relationships, proposing disentangled graph neural networks to model social relationships at a finer granularity.

Although GSR utilizes GNNs to effectively capture user-user social signals and user-item collaboration signals, its performance is still limited by noise [21] issue. Therefore, some research has begun to focus on the noise issue in GSR. DSL [26] achieves implicit denoising of social networks by adaptively aligning user similarities in both interaction and social domains. GDMSR [27] models the confidence of social relations and periodically reconstructs the social network to perform denoising. A model-agnostic denoising module for graph based social recommendation (MADM) [42] introduces a model-agnostic denoising module for GSR, which aims at generating a refined social structure and learning informative and noiseless user social representations. These efforts are based on preference-guided social network denoising paradigm [27], which has improved the robustness of the models to some extent. However, these works have not taken into account the noise in the interaction data, thus limiting the effectiveness of social network denoising and the performance of recommendation.

B. Graph Contrastive Learning for Recommendation

In recent years, contrastive learning has been successful in many fields (CV [43], natural language processing (NLP) [44]), and it effectively alleviates the problem of supervised signal sparsity especially on graph data, thus graph contrastive learning (GCL) [45] is increasingly being applied to graph-based recommedation including GSR. The core idea of GCL is to employ graph-augmentation (GA) or feature-augmentation (FA) and then minimize the loss such as InfoNCE [46] (illustrated later) to achieve the mutual information maximization [47] between the augmented samples to offer extra supervised signals.

In graph-based recommender systems, SGL [45] first utilizes GCL, which obtains two different views by randomly dropping edges/nodes on user-item interaction graph, and uses user/item embeddings in these two views for contrastive learning. Socially-aware self-supervised Tri-training framework

(SEPT) [18] jointly utilizes interaction graph and social network for data augmentation, and constructs the friend view and the sharing view for contrastive learning. DcRec [19] introduces a novel disentangled contrastive learning framework for social recommendation that leverages contrastive learning to transfer knowledge from the social domain to the collaborative domain. SimGCL [48] improves the data augmentation method of SGL by directly randomly perturbing the embeddings of users/items to generate positive samples for contrastive learning, and states that the key role is played by the contrastive learning loss function InfoNCE. ReACL [49] harnesses multiple views of user social relations and item commonality relations to form contrastive pairs and extract relationship-guided self-supervised signals.

GCL has significantly improved the performance of recommendation task, including GSR. However, the data augmentation methods used in GCL, have low interpretability and are uncontrollable, which can lead to the introduction of noise. Moreover, there may exist a gradient bias issue with InfoNCE during training process, which can impair the robustness of contrastive learning. Therefore, we can make further improvements to the data augmentation method and loss function to optimize the GCL paradigm and design a GCL framework that is more suitable for social recommendation scenarios, so as to better play the role of GCL in mitigating data sparsity and denoising.

III. Dual-Domain Collaborative Denoising For Social Recommendation

A. Problem Formulation

DCDSR aims to perform top-K item recommendations for the user set $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ from the item set $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$. To achieve this task, we formalize the input user-item interaction data and social trust data into a user-item interaction graph $\mathcal{G}_R$ and a social network $\mathcal{G}_S$.

B. Model Overview

As shown in Fig. 2, DCDSR consists of two main modules, i.e., structure-level collaborative denoising module and embedding-space collaborative denoising module. The structural-level collaborative denoising module is used to perform denoising on original social network and interaction graph to minimize the impact of structural noise on GNN learning user/item embeddings. Subsequently, the embedding-space collaborative denoising module contributes to alleviate the noise diffusion across interaction domain and social domain existing in structure-level collaborative denoising module. Eventually, the model is trained jointly with the Bayesian personalized ranking (BPR) [22] loss of recommendation task and AC-InfoNCE losses of contrastive learning task.

C. Structure-Level Collaborative Denoising

The goal of structure-level collaborative denoising is to evaluate the noise in the original social network $\mathcal{G}_S$ and interaction graph $\mathcal{G}_R$ and delete those noisy edges in the network/graph to obtain cleaner social network $\mathcal{G}'_S$ and interaction graph $\mathcal{G}'_R$.

1) Preference-Guided Social Network Denoising: In social recommendation, social network is used to assist the recommendation task which lies in there are lots of user relations and can provide extra supervised signals because social-related users tend to have consistent preferences according to the social homophily theory [5]. However, many studies have shown that there are lots of social relations with dissimilar preferences in the social network and these relations are harmful for recommendation task. In other words, only those social relations where social-pair users have high preference consistency can really facilitate the recommendation task. Therefore, we follow the principle of preference-guided social network denoising which requires evaluating the preference consistency of each social relation. Now we describe the process as follows.

First, we employ two-layer LightGCN [35] to perform multi-layer convolutional operations for the learnable user embedding $\mathbf{e}_u$ and item embedding $\mathbf{e}_i$ on the interaction graph (all denoised interaction graph but the original interaction graph in the first training epoch) to learn the user preference embedding and item embedding

$$\mathbf{h}_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{h}_i^{(l)}$$
$$\mathbf{h}_i^{(l+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{h}_u^{(l)} \tag{1}$$

where $\mathbf{h}_u^{(l)}$ and $\mathbf{h}_i^{(l)}$ represent the lth layer's embedding of user u and item i, the initial embeddings $\mathbf{h}_u^{(0)}$ and $\mathbf{h}_i^{(0)}$ correspond to $\mathbf{e}_u$ and $\mathbf{e}_i$, respectively. Finally, the user preference embedding $\mathbf{h}_u^r$ and item embedding $\mathbf{h}_i^r$ in interaction domain are obtained through average pooling operation on the embedding of all layers. Based on this, we compute the *preference consistency* (*PC*) between user u and user v of each social edge as follows:

$$PC(u, v) = \frac{1 + \cos(\mathbf{h}_u^r, \mathbf{h}_v^r)}{2} \cdot \exp\left(-\frac{||\mathbf{h}_u^r - \mathbf{h}_v^r||_2^2}{2\sigma^2}\right) \tag{2}$$

where $\mathbf{h}_u^r$ and $\mathbf{h}_v^r$ represent the preference embeddings of user u and user v, $\cos(\cdot)$ represents cosine similarity function. It is worth noting that we consider both direction consistency (cosine similarity) and spatial distance consistency (Gaussian kernel similarity) between preference embeddings to weigh the preference consistency more precisely, where σ controls the sensitivity of the kernel function to the Euclidean distance between $\mathbf{h}_u^r$ and $\mathbf{h}_v^r$, the value of whom would be discussed in Section IV-H. Eventually, the value of $PC(u, v)$ is within [0,1]. A higher preference consistency means that the social edge is more trustworthy, and instead it is more likely a noisy edge. Therefore, to filter potential noisy edges, we set the threshold β_s to remove the social edges with preference consistency scores less than the β_s and keep the remaining ones, which makes the social network retain important social information while enabling the network structure more lightweight. Then the denoised social network is denoted by $\mathcal{G}'_S$. The denoising process is expressed as follows:

$$\mathcal{G}'_S = \mathcal{G}_S \setminus \mathcal{N}_S,$$
$$\text{where } \mathcal{N}_S = \{(u, v) \in \mathcal{G}_S \mid PC(u, v) < \beta_s\}. \tag{3}$$

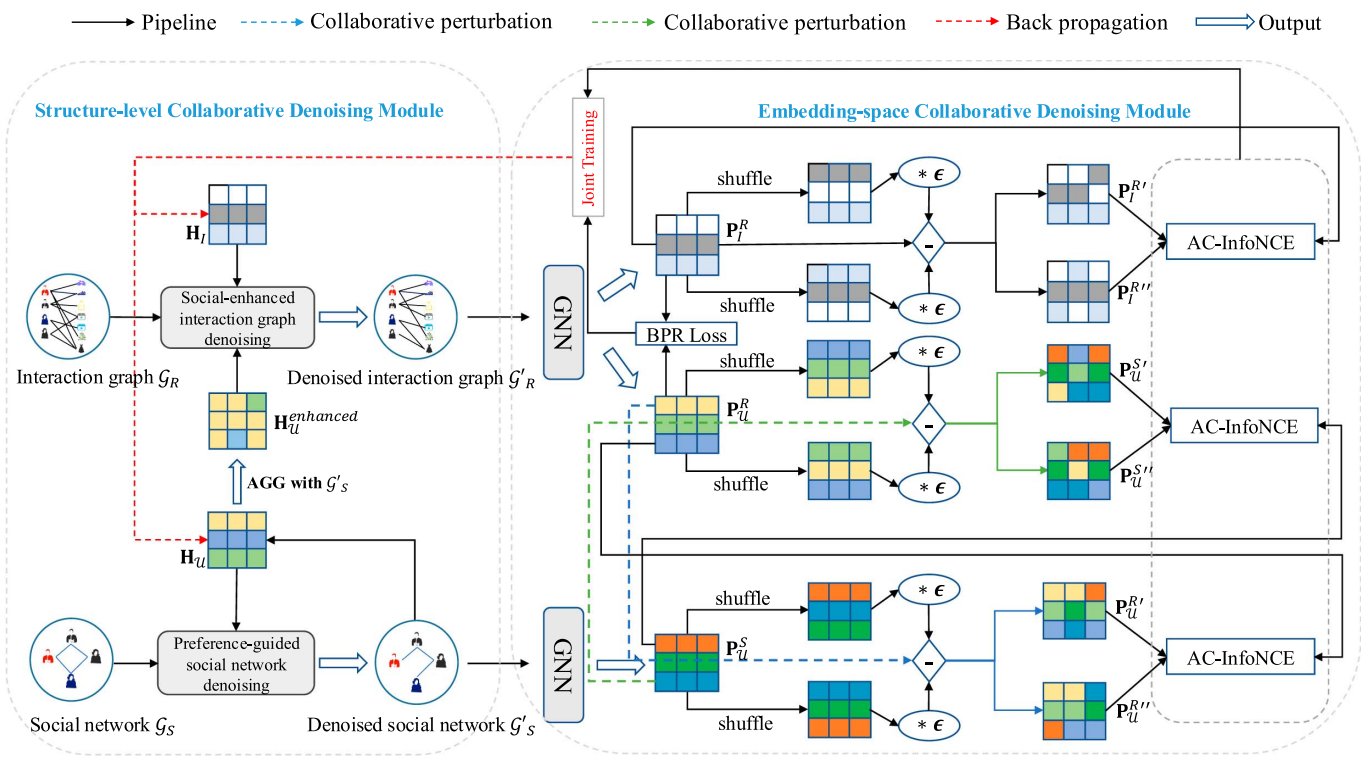

Fig. 2. Overview of DCDSR.

2) Social-Enhanced Interaction Graph Denoising: Since we use the knowledge of interaction domain learned by noisy interaction graph to guide social network denoising, the denoised social network still contains noise. Therefore, we need to further denoise the interaction graph to better guide the next round of social network denoising. We believe that the noise in the interaction graph comes from users' accident interactions or passive interactions due to social influence [8]. Therefore, we should consider whether the interacted items are truly compatible with the user. In fact, the formation of a reliable interaction graph follows the idea of user-based collaborative filtering [50], i.e., user's preference on certain item can be influenced by that of his/her similar users'. Hence, we draw support from the relatively cleaner social relations to derive user's true preference to maximally restore the true interaction graph. Specifically, we employ the $\text{AGG}_{\text{social}}$ function which means aggregating the preference embeddings of a user's one-order social neighbors in the denoised social network $\mathcal{G}'_S$ to obtain social-enhanced user preference embedding $\mathbf{h}_u^{\text{enhanced}}$

$$\begin{aligned}\mathbf{h}_u^{\text{enhanced}} &= \text{AGG}_{\text{social}}\left(\left\{\mathbf{h}_v^r \;\middle|\; v \in \mathcal{S}_u\right\}\right) \\ &= \sum_{v\in\mathcal{S}_u} \frac{1}{\sqrt{|\mathcal{S}_u|\cdot|\mathcal{S}_v|}}\mathbf{h}_v^r, \quad \forall u \in \mathcal{U}\end{aligned} \tag{4}$$

where $\mathcal{S}_u$ represents the set of user u's social neighbors. By doing this, the preferences within similar social groups are as consistent as possible, while the preferences between different social groups are as different as possible. After obtaining the social-enhanced user preference embedding in interaction domain, we jointly use item embedding to compute the *interaction compatibility (IC)* to determine whether the user with enhanced preference is compatible with their interacted items. Similarly, its calculation method is illustrated as

$$IC(u,i) = \frac{1+\cos\left(\mathbf{h}_u^{\text{enhanced}}, \mathbf{h}_i^r\right)}{2} \cdot \exp\left(-\frac{||\mathbf{h}_u^{\text{enhanced}} - \mathbf{h}_i^r||_2^2}{2\sigma^2}\right) \tag{5}$$

where the value of σ is samely setting as that in (2). Higher IC value means that user u is more compatible with item i, and vice versa. Thus, we set a threshold β_r, for each edge in the interaction graph, if its interaction compatibility is lower than the β_r, it may be a potentially noisy edge thus should be deleted. The denoising process is expressed as follows:

$$\begin{aligned}&\mathcal{G}'_R = \mathcal{G}_R \setminus \mathcal{N}_R, \\ &\text{where } \mathcal{N}_R = \left\{(u,\ i) \in \mathcal{G}_R \;\middle|\; IC(u,i) < \beta_r\right\}.\end{aligned} \tag{6}$$

Through the above process, we reduce the noise in the original interaction graph, which is beneficial for the next round of social network denoising and provides cleaner interaction data for the recommendation task, thereby improving the recommendation performance of the model.

Through the above two denoising segments, we achieve structure-level collaborative denoising which assures the denoising quality when there exists noise in both social network and interaction graph. At the same time, the social network truly plays a substantive role in providing supervisory signals for the interaction data. As the model iterates, $\mathcal{G}'_R$ and $\mathcal{G}'_S$ would get more and more trustworthy.

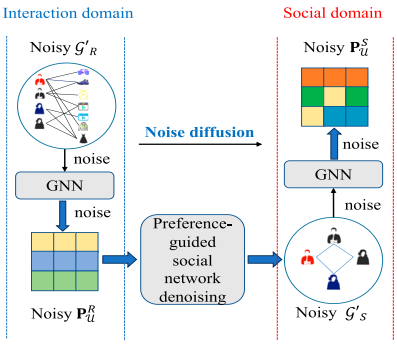

Fig. 3. Noise diffusion from interaction domain to social domain.

D. Embedding-Space Collaborative Denoising

1) Noise Cross-Domain Diffusion: After obtaining the denoised social network $\mathcal{G}'_S$ and interaction graph $\mathcal{G}'_R$, we use parallel GNN encoders [35] to learn the embeddings in interaction domain and social domain for the recommendation task

$$\begin{aligned} \mathbf{P}^R_{\mathcal{U}}, \mathbf{P}^R_{\mathcal{I}} &= \text{GNN}^R(\mathbf{E}_{\mathcal{U}}, \mathbf{E}_{\mathcal{I}}, \mathcal{G}'_R) \\ \mathbf{P}^S_{\mathcal{U}} &= \text{GNN}^S(\mathbf{E}_{\mathcal{U}}, \mathbf{E}_{\mathcal{I}}, \mathcal{G}'_S) \end{aligned} \tag{7}$$

where $\mathbf{P}^R_{\mathcal{U}}$ and $\mathbf{P}^R_{\mathcal{I}}$ are the user interaction embedding and item embedding matrices, $\mathbf{P}^S_{\mathcal{U}}$ is the user social embedding matrix, and $\mathbf{E}_{\mathcal{U}}$ and $\mathbf{E}_{\mathcal{I}}$ are the matrix formulations of $\mathbf{e}_u$ and $\mathbf{e}_i$.

However, $\mathbf{P}^R_{\mathcal{U}}$, $\mathbf{P}^R_{\mathcal{I}}$, and $\mathbf{P}^S_{\mathcal{U}}$ are containing noise. This problem arises from the following reason: The original interaction graph $\mathcal{G}_R$ and social network $\mathcal{G}_S$ contain noise while manual sturcture-level denoising process cannot completely eliminate the noise. This results in the phenomenon of noise cross-domain diffusion which would always accompany the structure denoising process. Specifically, the noise in interaction domain is diffused to social domain, and the noise in social domain is diffused to interaction domain. This leads to the embeddings learned by GNN based on the denoised interaction graph $\mathcal{G}'_R$ and social network $\mathcal{G}'_S$ still contain noise.

Taking the noise diffusion from interaction domain to social domain as an example which is illustrated in Fig. 3, in the preference-guided social network denoising process, embeddings $\mathbf{h}^r_u$, $\mathbf{h}^r_i$ learned by GNN on the denoised but still noisy interaction graph $\mathcal{G}'_R$ naturally contain semantic noise. Subsequently, based on these embeddings, (2) and (3) are used for social network denoising, resulting in edges in $\mathcal{G}_S$ may be ignored for deletion or incorrectly deleted, thus the denoised social network $\mathcal{G}'_S$ still contains noise. Consequently, the $\mathbf{P}^S_{\mathcal{U}}$ learned based on $\mathcal{G}'_S$ contains noise. In summary, the noise in the interaction domain is diffused to the social domain. Similarly, social-enhanced interaction graph denoising process also diffuses the noise from the social domain to the interaction domain, causing $\mathbf{P}^R_{\mathcal{U}}$ and $\mathbf{P}^R_{\mathcal{I}}$ to contain noise.

Therefore, we further perform denoising in the embedding space. We desire to restore the noise cross-domain diffusion process from the perspective of embedding space, then try to simulate the user and item embeddings uncontaminated by diffused noise in the form of dual-domain collaborative embedding perturbation, and finally perform embedding denoising with contrastive learning.

2) Embedding Collaborative Perturbation: Specifically, above diffusion process we have illustrated can be simplified to that $\mathbf{P}^R_{\mathcal{U}}$ symbolizing information of interaction domain and $\mathbf{P}^S_{\mathcal{U}}$ symbolizing information of social domain perturb each other. Therefore, for user social embeddings, we generate a kind of noise that satisfies the knowledge distribution of interaction domain

$$\Delta_S = \text{sign}(\mathbf{P}^S_{\mathcal{U}}) \odot \frac{f_{\text{shuffle}}(\mathbf{P}^R_{\mathcal{U}})}{||f_{\text{shuffle}}(\mathbf{P}^R_{\mathcal{U}})||_2} \tag{8}$$

where $\text{sign}(\cdot)$ represents the symbol function which requires that $\mathbf{P}^S_{\mathcal{U}}$ and Δ_S are in the same hyperoctant, $\odot$ is the Hadama product, f_{shuffle} [51] means shuffling the rows of the $\mathbf{P}^R_{\mathcal{U}}$, aiming to simulate the uncertain false-positive connections among users generated in preference-guided social network denoising.

Subsequently, this noise is subtracted from the noisy user social embedding to simulate user social embedding uncontaminated by the noise from interaction domain as follows:

$$\mathbf{P}^{S\prime}_{\mathcal{U}} = \mathbf{P}^S_{\mathcal{U}} - \epsilon * \Delta_S \tag{9}$$

where the magnitude of final perturbation noise is numerically equivalent to ϵ.

Similarly, for user interaction embeddings, we generate a kind of noise that satisfies the knowledge distribution of the social domain. Given that the social-enhanced interaction graph denoising process diffuses the noise from social domain to interaction domain, we subtract this noise from user interaction embedding to maximally simulate the raw user interaction embedding uncontaminated by the noise from social domain. It can be described as

$$\begin{aligned} \Delta_R &= \text{sign}(\mathbf{P}^R_{\mathcal{U}}) \odot \frac{f_{\text{shuffle}}(\mathbf{P}^S_{\mathcal{U}})}{||f_{\text{shuffle}}(\mathbf{P}^S_{\mathcal{U}})||_2} \\ \mathbf{P}^{R\prime}_{\mathcal{U}} &= \mathbf{P}^R_{\mathcal{U}} - \epsilon * \Delta_R. \end{aligned} \tag{10}$$

Through the above approach, we achieve dual-domain embedding collaborative perturbation, which explicitly defines the source and semantics of the perturbation noise. It can, to a certain extent, mitigate the noise cross-domain diffusion, thereby enabling the perturbed embeddings to maximally simulate the raw embeddings unaffected by noise diffusion.

In addition, for item embeddings, we use a similar method to generate noise that satisfy the item knowledge distribution, trying to mitigate the information propagation between items due to the noise to simulate the item embeddings unaffected by noise

$$\mathbf{P}^{R\prime}_{\mathcal{I}} = \mathbf{P}^R_{\mathcal{I}} - \epsilon * \text{sign}(\mathbf{P}^R_{\mathcal{I}}) \odot \frac{f_{\text{shuffle}}(\mathbf{P}^R_{\mathcal{I}})}{||f_{\text{shuffle}}(\mathbf{P}^R_{\mathcal{I}})||_2}. \tag{11}$$

Finally, the above three embedding perturbation processes need to be repeated twice respectively to obtain positive samples $\mathbf{P}^{S\prime}_{\mathcal{U}}$, $\mathbf{P}^{S\prime\prime}_{\mathcal{U}}$ of user social embeddings, $\mathbf{P}^{R\prime}_{\mathcal{U}}$, $\mathbf{P}^{R\prime\prime}_{\mathcal{U}}$ of user interaction embeddings and $\mathbf{P}^{R\prime}_{\mathcal{I}}$, $\mathbf{P}^{R\prime\prime}_{\mathcal{I}}$ of item embeddings for contrastive learning.

3) Contrastive Learning for Embedding-Space Denoising: After performing embedding perturbation and obtaining positive samples, we perform contrastive learning to denoise the original embeddings. Especially, we propose a novel contrastive learning loss function Anchor-InfoNCE (AC-InfoNCE) to improve InfoNCE [46], which set the original embedding as an anchor and simultaneously pull the anchor closer to its two positive samples together in the embedding space, while pushing apart embeddings from different users/items. InfoNCE ($\mathcal{L}_{\mathrm{I}}$) and AC-InfoNCE ($\mathcal{L}_{\mathrm{AC}}$) are illustrated as (12)

$$\begin{aligned}\mathcal{L}_{\mathrm{I}} &= \sum_{i\in\mathcal{B}} -\log\left(\frac{\exp\left(s(\mathbf{h}_i', \mathbf{h}_i'')/\tau\right)}{\sum_{j\in\mathcal{B}}\exp\left(s(\mathbf{h}_i', \mathbf{h}_j'')/\tau\right)}\right)\\ \mathcal{L}_{\mathrm{AC}} &= \sum_{i\in\mathcal{B}} -\log\left(\frac{\exp\left((s(\mathbf{h}_i, \mathbf{h}_i') + s(\mathbf{h}_i, \mathbf{h}_i''))/2\tau\right)}{\sum_{j\in\mathcal{B}}\exp\left(s(\mathbf{h}_i', \mathbf{h}_j'')/\tau\right)}\right)\end{aligned} \tag{12}$$

where $\mathbf{h}_i$ denotes the original embedding of user/item i, $\mathbf{h}_i'$, $\mathbf{h}_i''$ denotes two different augmented positive samples of $\mathbf{h}_i$, $\mathbf{h}_j''$ denotes the second positive sample of user/item j, $s(\cdot)$ and τ represent similarity calculation operation which is cosine similarity here and temperature coefficient, respectively.

AC-InfoNCE is proposed to allievate the gradient bias issue due to the uncertainty of embedding perturbation and the distribution of negative samples. Now we explain this issue and how AC-InfoNCE can effectively deal with it by combining numerical analysis (13)–(16) with geometrical representation (Fig. 4).

Equations (13) and (14) respectively represent the gradient calculation process of InfoNCE loss with respect to $\mathbf{h}_i'$ and $\mathbf{h}_i''$ (i.e., $-\nabla_i'$ and $-\nabla_i''$), where $\tilde{\mathbf{h}}_i'$ represent the L_2 norm of $\mathbf{h}_i'$. We can see the fact that when $s(\mathbf{h}_i', \mathbf{h}_i'')$ is very large while $s(\mathbf{h}_i', \mathbf{h}_j'')$ and $s(\mathbf{h}_i'', \mathbf{h}_j'')$ are very small, ∇_i' and ∇_i'' both approach 0, which is the driving force behind gradient update of InfoNCE. However, due to the uncertainty of data augmentation and the distribution of negative samples, there may be cases where $\mathbf{h}_i'$ has a high similarity with certain negative example $\mathbf{h}_j''$. The negative gradient may exacerbate the tendency for $\mathbf{h}_i'$ to approach $\mathbf{h}_i''$ to a certain extent, while $\mathbf{h}_i''$ still updates toward $\mathbf{h}_i'$ with a small scope, resulting in the final update direction of $\mathbf{h}_i$ (∇_i) being completely dominated by ∇_i', in other words, it greatly biases $\mathbf{h}_i$ towards $\mathbf{h}_i''$. This is the so-called gradient bias issue, which would eventually lead to that the robustness of the finally optimized embedding $\mathbf{h}_i^{\mathrm{align}}$ is vulnerable to embedding perturbation

$$\begin{aligned}\frac{\partial\mathcal{L}_{\mathrm{I}}}{\partial\mathbf{h}_i'} &= \frac{1}{\tau}\left(\frac{\sum_{j\in\mathcal{B}}\exp(s(\mathbf{h}_i', \mathbf{h}_j'')/\tau))\cdot\frac{\partial s(\mathbf{h}_i', \mathbf{h}_j'')}{\partial\mathbf{h}_i'}}{\sum_{j\in\mathcal{B}}\exp(s(\mathbf{h}_i', \mathbf{h}_j'')/\tau))} - \frac{\partial s(\mathbf{h}_i', \mathbf{h}_i'')}{\partial\mathbf{h}_i'}\right)\\ &= \frac{1}{\tau}\left(\sum_{j\in\mathcal{B}}\frac{\exp(s(\mathbf{h}_i', \mathbf{h}_j'')/\tau)}{\sum_{k\in\mathcal{B}}\exp(s(\mathbf{h}_i', \mathbf{h}_k'')/\tau)}\frac{\mathbf{h}_j''}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_j''} - \frac{\mathbf{h}_i''}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_i''}\right)\\ &= \frac{1}{\tau}\left(\mathbb{E}_{j\sim p(j|i)}\left[\frac{\mathbf{h}_j''}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_j''}\right] - \frac{\mathbf{h}_i''}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_i''}\right)\\ p(j|i) &= \frac{\exp(s(\mathbf{h}_i', \mathbf{h}_j'')/\tau)}{\sum k\in\mathcal{B}\exp(s(\mathbf{h}_i', \mathbf{h}_k'')/\tau)}\end{aligned} \tag{13}$$

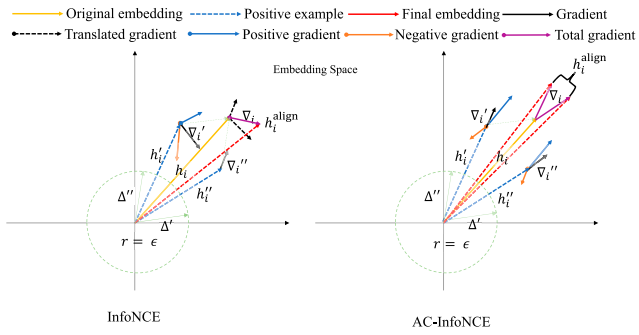

Fig. 4. Process of embedding optimizing with InfoNCE and AC-InfoNCE.

$$\frac{\partial\mathcal{L}_{\mathrm{I}}}{\partial\mathbf{h}_i''} = \frac{1}{\tau}\left(p(i|i)\left[\frac{\mathbf{h}_i'}{\tilde{\mathbf{h}}_i''\tilde{\mathbf{h}}_i'}\right] - \frac{\mathbf{h}_i'}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_i''}\right). \tag{14}$$

Based on this, we design the AC-InfoNCE, which sets $\mathbf{h}_i$ as an anchor point and prompts $\mathbf{h}_i'$ and $\mathbf{h}_i''$ to converge to alignment with $\mathbf{h}_i$ at the same time. According to (15), we can see that ∇_i essentially lies within the intermediate direction of $\mathbf{h}_i$ and $\mathbf{h}_i'$ and the intermediate direction of $\mathbf{h}_i$ and $\mathbf{h}_i''$. Even with the presence of negative sample $\mathbf{h}_j''$ which makes $s(\mathbf{h}_i', \mathbf{h}_j'')$ high, it does not greatly affect the outcome. After embedding optimizing, $\mathbf{h}_i^{\mathrm{align}}$ is obtained, which has only a minor deviation of $\mathbf{h}_i$ rather than significantly approaching $\mathbf{h}_i'$ or $\mathbf{h}_i''$, thereby ensuring that the final embeddings are robust. We can also draw this inclusion from (16) that both $\mathbf{h}_i'$ and $\mathbf{h}_i''$ are moving closer to $\mathbf{h}_i$, but ultimately ∇_i' and ∇_i'' would not cross the direction of $\mathbf{h}_i$. This is the reason why $\mathbf{h}_i$ does not shift significantly in the end

$$\begin{aligned}\frac{\partial\mathcal{L}_{\mathrm{AC}}}{\partial\mathbf{h}_i} &= -\frac{1}{2\tau}\left(\frac{\mathbf{h}_i' + \mathbf{h}_i}{\tilde{\mathbf{h}}_i\tilde{\mathbf{h}}_i'} + \frac{\mathbf{h}_i'' + \mathbf{h}_i}{\tilde{\mathbf{h}}_i\tilde{\mathbf{h}}_i''}\right)\\ &\quad + \frac{1}{2\tau}\left(\mathbb{E}_j\sim p(j|i)\left[\frac{2\mathbf{h}_j''}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_j''}\right]\right)\end{aligned} \tag{15}$$

$$\frac{\partial\mathcal{L}_{\mathrm{AC}}}{\partial\mathbf{h}_i'} = -\frac{1}{2\tau}\left(\frac{\mathbf{h}_i}{\tilde{\mathbf{h}}_i\tilde{\mathbf{h}}_i'} - \mathbb{E}_j\sim p(j|i)\left[\frac{2\mathbf{h}_j''}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_j''}\right]\right)$$

$$\frac{\partial\mathcal{L}_{\mathrm{AC}}}{\partial\mathbf{h}_i''} = -\frac{1}{2\tau}\left(\frac{\mathbf{h}_i}{\tilde{\mathbf{h}}_i\tilde{\mathbf{h}}_i''} - p(i|i)\left[\frac{2\mathbf{h}_i'}{\tilde{\mathbf{h}}_i'\tilde{\mathbf{h}}_i''}\right]\right). \tag{16}$$

Moreover, AC-InfoNCE incorporates the original embedding into contrastive learning, which is equivalent to treating it as a positive sample, expanding the number of positive samples for contrastive learning, thus alleviating the data sparsity problem and further realizing the denoising effect.

E. Joint Training

DCDSR utilizes multitask joint training to tune the model. The entire model incorporates the losses of recommendation task and contrastive learning task to guide the model in learning the user and item embeddings. Given the user interaction embedding and item embedding, we predict the likelihood of user u interacting with item i, formulated as

$$\hat{y}_{ui} = \mathbf{p}_u^{r\mathrm{T}}\mathbf{p}_i. \tag{17}$$

Based on this, we minimize the following BPR [22] loss to optimize the main recommendation task

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i,j)\in O} -\log \sigma\left(\hat{y}_{u,i} - \hat{y}_{u,j}\right) \tag{18}$$

where $\sigma(\cdot)$ is the sigmoid function, $\mathcal{O} = \{(u,i,j) \mid (u,i) \in \mathcal{O}^+, (u,j) \in \mathcal{O}^-\}$. Here, $\mathcal{O}^+$ is the observed pairwise training data, and $\mathcal{O}^-$ denotes the unobserved interactions. The contrastive learning task consists of CL losses $\mathcal{L}^r_{\text{CL}}$ for user interaction embeddings, $\mathcal{L}^s_{\text{CL}}$ for user social embeddings, and $\mathcal{L}^i_{\text{CL}}$ for item embeddings, as follows:

$$\begin{aligned}
\mathcal{L}^r_{\text{CL}} &= \sum_{u\in\mathcal{B}} -\log\left(\frac{\exp\left((s(\mathbf{p}^r_u, \mathbf{p}^{r\prime}_u) + s(\mathbf{p}^r_u, \mathbf{p}^{r\prime\prime}_u))/2\tau\right)}{\sum_{v\in\mathcal{B}}\exp\left(s(\mathbf{p}^{r\prime}_u, \mathbf{p}^{r\prime\prime}_v)/\tau\right)}\right) \\
\mathcal{L}^s_{\text{CL}} &= \sum_{u\in\mathcal{B}} -\log\left(\frac{\exp\left((s(\mathbf{p}^s_u, \mathbf{p}^{s\prime}_u) + s(\mathbf{p}^s_u, \mathbf{p}^{s\prime\prime}_u))/2\tau\right)}{\sum_{v\in\mathcal{B}}\exp\left(s(\mathbf{p}^{s\prime}_u, \mathbf{p}^{s\prime\prime}_v)/\tau\right)}\right) \\
\mathcal{L}^i_{\text{CL}} &= \sum_{i\in\mathcal{B}} -\log\left(\frac{\exp\left((s(\mathbf{p}^r_i, \mathbf{p}^{r\prime}_i) + s(\mathbf{p}^r_i, \mathbf{p}^{r\prime\prime}_i))/2\tau\right)}{\sum_{j\in\mathcal{B}}\exp\left(s(\mathbf{p}^{r\prime}_i, \mathbf{p}^{r\prime\prime}_j)/\tau\right)}\right).
\end{aligned} \tag{19}$$

Finally, the loss of the recommendation task and the losses of the contrastive learning task are combined to obtain the total loss of DCDSR as follows:

$$\begin{aligned}
\mathcal{L}_{\text{DCDSR}} = \mathcal{L}_{\text{BPR}} &+ \lambda_1\mathcal{L}^r_{\text{CL}} + \lambda_2\,\mathcal{L}^s_{\text{CL}} + \lambda_3\,\mathcal{L}^i_{\text{CL}} \\
&+ \lambda_{\text{reg}}\left(||\mathbf{E}_{\mathcal{U}}||^2_F + ||\mathbf{E}_{\mathcal{I}}||^2_F\right)
\end{aligned} \tag{20}$$

where λ_1, λ_2, and λ_3 are the coefficients controlling the effects of three CL loss respectively, and λ_{reg} represents the regularization weight for the user/item embedding. The whole training process of DCDSR is shown in Algorithm 1.

Algorithm 1: The overall training process of DCDSR.

Input: The original interaction graph $\mathcal{G}_R$ and social network $\mathcal{G}_S$, hyperparameters β_r, β_s, λ_1, λ_2, λ_3;
Output: The user embeddings $\mathbf{E}_{\mathcal{U}}$ and item embeddings $\mathbf{E}_{\mathcal{I}}$;

Initialize parameters $\mathbf{E}_{\mathcal{U}}$ and $\mathbf{E}_{\mathcal{I}}$ with the default Xavier distribution;
while DCDSR not converge **do**
  Reconstruct a denoised social network $\mathcal{G}'_S$ based on original social network $\mathcal{G}_S$ with Eqs. (1)-(3);
  Reconstruct a denoised interaction graph $\mathcal{G}'_R$ based on denoised social network $\mathcal{G}'_S$ with Eqs. (4)-(6);
  Sample a mini-batch of user $\mathcal{B} \in \mathcal{U}$;
  for $u \in \mathcal{B}$ **do**
    Obtain user social embeddings $\mathbf{P}^R_{\mathcal{U}}$, item embeddings $\mathbf{P}^R_{\mathcal{I}}$ and user social embeddings $\mathbf{P}^S_{\mathcal{U}}$ for each user u and item i based on $\mathcal{G}'_R$ and $\mathcal{G}'_S$ with Eq. (7);
    Calculate BPR loss $\mathcal{L}_{\text{BPR}}$ with Eqs. (17)-(18);
    Obtain perturbed embeddings $\mathbf{P}^{R\prime}_{\mathcal{U}}$, $\mathbf{P}^{R\prime\prime}_{\mathcal{U}}$, $\mathbf{P}^{S\prime}_{\mathcal{U}}$, $\mathbf{P}^{S\prime\prime}_{\mathcal{U}}$, $\mathbf{P}^{R\prime}_{\mathcal{I}}$, and $\mathbf{P}^{R\prime\prime}_{\mathcal{I}}$ based on $\mathbf{P}^R_{\mathcal{U}}$, $\mathbf{P}^S_{\mathcal{U}}$, and $\mathbf{P}^R_{\mathcal{I}}$ with Eqs. (8)-(11);
    Calculate AC-InfoNCE losses $\mathcal{L}^r_{\text{CL}}$,$\mathcal{L}^s_{\text{CL}}$, and $\mathcal{L}^i_{\text{CL}}$ for contrastive learning, according to Eq. (19);
    Calculate joint learning loss $\mathcal{L}_{\text{DCDSR}}$ with Eq. (20);
    Backpropagation and update parameters $\mathbf{E}_{\mathcal{U}}$ and $\mathbf{E}_{\mathcal{I}}$ with $\mathcal{L}_{\text{DCDSR}}$;
  end for;
end while;
return parameters $\mathbf{E}_{\mathcal{U}}$ and $\mathbf{E}_{\mathcal{I}}$;

F. Discussion

1) Novelty and Difference: In this section, we compare DCDSR with various denoising GSR and GCL-based GSR models to highlight the novelty of DCDSR.

First, DCDSR is a pioneering denoising GSR model that collaboratively performs denoising on social domain and interaction domain, moreover, it executes denoising at both the structure level and the embedding space. In contrast, DSL [26] only performs social domain denoising through adaptive alignment at the embedding space. GDMSR [27] jointly denoises the social domain at the structure level and embedding space. It can be seen that these models do not consider denoising the interaction domain noise, and some of them only perform denoising at a single perspective.

In addition, the embedding-space collaborative denoising module of DCDSR incorporates graph contrastive learning. In terms of data augmentation method, unlike the GA used in SEPT [18] and DcRec [19], DCDSR adopts FA, which is more efficient and avoids the loss of important graph structure information during the GA process. At the same time, compared with general FA method [48] which performs embedding perturbation at each layer of GNN, DCDSR only performs embedding perturbation at the last layer of GNN. Furthermore, the noise used for embedding perturbation in DCDSR is sampled from the interaction/social domain knowledge distribution, rather than the random Gaussian noise [48]. This approach restores the process of noise cross-domain diffusion in social recommendation and achieves embedding-space denoising with better explainability.

2) Complexity Analysis: In this section, we analyze the theoretical temporal complexity of DCDSR. Table II provides a comparison between DCDSR and some state-of-the-art graph-based recommendation models in terms of time complexity. Some models introduce GCL while others are designed for denoising task. Therefore, temporal complexity analysis of GNN operation and structure-level denoising (SD) are given and whether they perform emebdding-space denoising (ED) is provided. The numbers of edges and vertices of the interaction graph are $|E|$ and $|V|$, while the number of edges in the social network is represented by $|S|$. In the encoder, L represents the number of GNN layers, d denotes the embedding size, s is the number of epochs required for training, B indicates the batch size, and ρ corresponds to the edge retention rate in GA. Furthermore, A_f, As represent the friend and share views engendered by the SEPT method, respectively, while H_s, H_j, H_p are the triad of convolutional channels for social, joint,

TABLE II
TEMPORAL COMPLEXITY ANALYSIS

	GNN	GD	ED
NGCF	$O((\|E\|+\|V\|)Lds\|E\|/B)$	×	×
LightGCN	$O(\|E\|Lds\|E\|/B)$	×	×
DiffNet	$O((\|S\|L+\|E\|)ds\|E\|/B)$	×	×
S^2-MHCN	$O\left((\|H_s\|+\|H_j\|+\|H_p\|+\|E\|)\,Lds\|E\|/B\right)$	×	×
SEPT	$O\left((\|A_f\|+\|A_s\|+\|E\|+\rho_1(\|E\cup S\|))\,Lds\|E\|/B\right)$	×	×
DESIGN	$O(2(\|E\|+\|S\|)Lds\|E\|/B)$	×	×
DcRec	$O\left(((1+2\rho_1)\,\|E\|+2\rho_2\|S\|)\,Lds\|E\|/B\right)$	×	×
DSL	$O((\|E\|+\|S\|)Lds\|E\|/B)$	×	✓
GDMSR	$O((\|E\|+\|S\|)Lds\|E\|/B)$	$O(\|S\|s/D)$	×
DCDSR	$O((\|E\|+\|S\|)Lds\|E\|/B)$	$O((\|E\|+\|S\|)s)$	✓

TABLE III
STATISTICS OF DATASE TS

Dataset	#User	#Item	#Interaction	#Relation
Douban-Book	13 024	22 347	792 062	169 150
Yelp	19 539	21 266	450 884	363 672
Douban-movie	2,848	39 586	894 887	35 770
Ciao	7,335	17 867	140 628	111 679
Yelp2	32 827	59 972	598 121	964 510
Douban	2,669	15 940	535 210	32 705

and purchase within MHCN. D denotes to a social network reconstruction every D epochs in GDMSR. For DCDSR, the time complexity for one iteration is consist of GNN encoding which consumes $O((|E|+|S|)Ld*|E|/B)$ and SD for social network and interaction graph and which totally consume $O(|E|+|S|)$. Therefore, after s iterations, the total time complexity becomes $O((|E|+|S|)Lds|E|/B+(|E|+|S|)s)$.

IV. EXPERIMENTS

We conduct experiments on three real-world datasets to validate the superiority of DCDSR compared with some state-of-the-art models and answer the following questions.

1) RQ1: How does DCDSR perform compared with these GSR and denoising social recommendation models?
2) RQ2: Do each key component and method of DCDSR contribute to the model?
3) RQ3: How about the noise-resistance of DCDSR compared with traditional GSR and denoising social recommendation models in the face of different proportions of noise attack?
4) RQ4: How does the performance of DCDSR change as the values of the model hyperparameters change?

A. Datasets

In this article, we adopt six benchmark datasets Douban-book [18], Yelp [52], Douban-movie [17], Ciao [27], Yelp2 [27], and Douban [27] to evaluate DCDSR, all of which are publicly available and cover different domains with various data sizes and densities. Table III lists the details of these datasets. To be consistent with previous work, the interaction data from all the datasets are transformed into implicit feedback [18], [32], [35]. Following the partition strategy of datasets involved [32], we first divided each dataset into the training set and the test set in a ratio of 8:2, then we randomly selected 10% of interaction data from the training set as validation set to tune hyper-parameters, ensuring that the test set remains untouched for the final evaluation.

B. Baselines

To validate the effectiveness of the proposed model and answer the questions raised above, we compare DCDSR with the following baselines.

1) *NGCF* [32]: This model utilizes standard GCN to explore complex relationships between users and items, boosting neighbor propagation through element-wise multiplication.
2) *LightGCN* [35]: This model streamlines GCN for recommendation by eliminating nonlinear activations, feature transformations, and self-connections.
3) *DiffNet* [9]: This model portrays social information diffusion process in social networks for GSR.
4) *MHCN* [17]: This model uses multichannel hypergraph convolutional networks to model complex user relationships for GSR.
5) *DESIGN* [21]: This GSR method constructs models of SocialGCN, RatingGCN, and TwinGCN to respectively model users and items, and employs knowledge distillation to collaboratively boost recommendation performance.
6) *S^2-MHCN* [17]: This model introduces GCL into MHCN.
7) *SEPT* [18]: This leading GSR model constructs various social network perspectives for GCL to encode multiple user embeddings, providing supervisory signals to the main task.
8) DcRec [19]: This GSR model constructs augmented graphs to learn different perspectives of node embeddings from both interaction and social domains to provide supervision signals.
9) *DSL* [26]: This method is a denoising GSR model that enables denoised cross-view alignment for similarities between different users.
10) *GDMSR* [27]: This method is a denoising framework for GSR which 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.

C. Parameter Settings

We implement our DCDSR in PyTorch[1]. All experiments are conducted on a single Linux server with Xeon(R) Gold 6133 CPU, 20G RAM, and 1 NVIDIA GeForce RTX 3090 GPU. All models are optimized using Adam [53] with parameters initialized by the Xavier method [54]. For fair comparisons, the embedding size is set to 50 for all the baselines except GMDSR, where we set it to 8 according to the original article. The mini-batch size is fixed to 2048. Following the original articles' suggested settings, we perform a grid search to select the optimal hyper-parameters for all models. The layer number

[1]https://pytorch.org/

TABLE IV
OVERALL PERFORMANCE COMPARISON

Datasets	Metrics	NGCF	LightGCN	DiffNet	MHCN	DESIGN	S^2-MHCN	SEPT	DcRec	DSL	**DCDSR**	Improve. %
Douban-Book	R@10	0.0813	0.0900	0.0845	0.1011	0.0945	0.1063	0.1058	0.1085	0.1005	**0.1252***	+15.39%
	N@10	0.1015	0.1238	0.1068	0.1154	0.1255	0.1385	0.1373	0.1404	0.1317	**0.1689***	+16.64%
	R@20	0.1261	0.1376	0.1272	0.1504	0.1416	0.1559	0.1545	0.1597	0.1503	**0.1794***	+12.34%
	N@20	0.1112	0.1302	0.1144	0.1408	0.1300	0.1461	0.1432	0.1466	0.1376	**0.1726***	+17.18%
Yelp	R@10	0.0554	0.0643	0.0535	0.0687	0.0703	0.0789	0.0746	0.0761	0.0727	**0.0871***	+10.39%
	N@10	0.0419	0.0512	0.0406	0.0531	0.0535	0.0606	0.0578	0.0585	0.0551	**0.0670***	+10.56%
	R@20	0.0903	0.1048	0.0873	0.1087	0.1113	0.1207	0.1126	0.1205	0.1164	**0.1346***	+11.52%
	N@20	0.0533	0.0641	0.0516	0.0657	0.0662	0.0740	0.0676	0.0723	0.0687	**0.0820***	+10.81%
Douban-Movie	R@10	0.0507	0.0533	0.0473	0.0541	0.05513	0.0554	0.0574	0.0573	0.0567	**0.0661***	+15.16%
	N@10	0.2315	0.2491	0.2049	0.2409	0.2429	0.2572	0.2672	0.2508	0.2470	**0.2862***	+7.11%
	R@20	0.0830	0.0883	0.0816	0.0909	0.0934	0.0926	0.0932	0.0947	0.0923	**0.1116***	+17.85%
	N@20	0.2161	0.2331	0.1944	0.2251	0.2324	0.2375	0.2466	0.2333	0.2287	**0.2663***	+7.99%

Note: The model that performs best on each dataset and metric is bolded, whereas the best baseline is underlined. 'Improve. %' indicates the relative improvement of DCDSR over the best baseline. R@10, N@10, R@20 and N@20 represent Recall@10, NDCG@10, Recall@20 and NDCG@20, respectively. Symbol * denotes that the improvement is significant with a p-value < 0.05 based on a two-tailed paired t-test.

TABLE V
PERFORMANCE COMPARISON BETWEEN GDMSR AND DCDSR

Datasets	Ciao		Yelp2		Douban	
Method	Recall@3	NDCG@3	Recall@3	NDCG@3	Recall@3	NDCG@3
GDMSR$_D$	0.1244	0.2821	0.3291	0.6102	0.2540	0.5297
GDMSR$_M$	0.1138	0.2632	0.3405	0.6434	0.3496	0.6137
DCDSR	**0.1426*(+14.63%)**	**0.3084*(+9.32%)**	**0.4517*(+32.66%)**	**0.6471*(+0.57%)**	**0.4561*(+30.46%)**	**0.7003*(+14.11%)**

Note: GDMSR$_D$ and GDMSR$_M$ represent employing GDMSR based on DiffNet++ [10] and MHCN, respectively. The best performance of all methods on each dataset is bolded, whereas the better performance of two variants of GDMSR on each dataset is underlined.

of GNNs for all models is tuned in [1, 2, 3]. For DCDSR, we set the layer number L as 2. We tune the structure-level collaborative denoising threshold β_s and β_r in the range of [0.5, 0.6, 0.7, 0.8, 0.9] and [0.3, 0.35, 0.4, 0.45, 0.5], respectively. The proportional coefficients of contrastive losses in three different views are all tuned in the range of [0.1, 0.2, 0.3, 0.4]. The temperature parameter τ for AC-InfoNCE and the magnitude of perturbation noise ϵ are tuned within [0.05, 0.1, 0.2, 0.3, 0.5, 1.0] and [0.01, 0.05, 0.1, 0.2, 0.5, 1.0], respectively. The Gaussian kernel parameter σ is tuned in [1, 5, 10, 20, 40, 60, 100].

D. Evaluation Protocols

To assess the performance of Top-N recommendation, in regular baselines, we employ two commonly used evaluation metrics Recall and NDCG which are computed by the all-ranking protocol [35], [45]. And in GDMSR, real-plus-N [27] is adopted to compute these metrics which randomly selects 100 uninteracted items for each user and ranks them together with the positive test samples.

E. Performance Comparison (RQ1)

As Tables IV and V show, Overall, on each datasets, DCDSR achieves the best performance on every metric, achieving an average of more than 10% improvement over all baseline models. Specifically, on Douban-Book, Yelp, and Douban-Movie, DCDSR achieves 12.27%, 15.95%, and 10.69% improvement on Recall@20, and 17.74%, 10.63%, and 10.54% improvement on NDCG@20. On Ciao, Yelp2 and Douban, DCDSR achieves 14.63%, 32.66%, and 30.46% improvement on Recall@3, and 9.32%, 0.57%, and 14.11% improvement on NDCG@3 compared with GMDSR. This is a strong proof of the significant effectiveness of DCDSR and its great universality to different data environments. We attribute these improvements to the following reasons:

1) Adhering to the dual-domain collaborative denoising paradigm ensures that the social network denoising task is less vulnerable to the noise in original interaction data. This approach not only enhances the quality of social network but also optimizes interaction data, thereby directly facilitating recommendation task. Building on preference-guided social network denoising, the denoised social network further serves to purify the interaction graph through knowledge enhancement, thus better exploiting the supervisory role of social network in recommendation task.

2) The embedding-space collaborative denoising module to some extent restores the noise cross-domain diffusion process. It simulates the raw embeddings unaffected by noise cross-domain diffusion through dual-domain embedding collaborative perturbations. Finally, these perturbed embeddings, along with the original embeddings, are used for contrastive learning, which denoises the embeddings and successfully mitigates the noise cross-domain diffusion issue. In addition, user social embeddings and user interaction embeddings are united by the contrastive learning task thus providing additional supervised signals for the recommendation task,

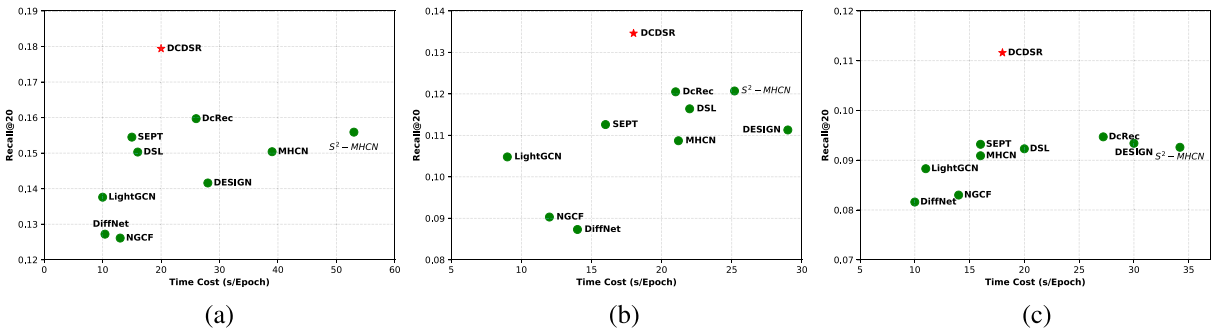

Fig. 5. Combined comparison of recommendation performance (y-axis, the higher the better) and training efficiency (x-axis, the lower the better) of DCDSR with other baseline methods. (a) Douban-book. (b) Yelp. (c) Douban-movie.

bridging the information barriers between social and interaction domains.

Combining all these baselines, GCL significantly improves the performance of GSR models. Taking DcRec as an example, it improves the performance of GSR by decoupling the user behavior into the representation of collaborative and social domains, and adopting the decoupled contrastive learning mechanism to transfer knowledge between the two domains. However, it has the following drawbacks compared with DCDSR. (1) It has not considered the noise issue in social recommendation. (2) It generates different views for contrastive learning by random graph augmentation, which may lose important edges in the original graph/network structure, thus leading to degradation of recommendation performance.

Further, we compare DSL and GDMSR with DCDSR, respectively. As it can be seen, DCDSR significantly outperforms DSL on all datasets due to the fact that DSL implicitly reduces noise by aligning the similarity of user-user pairs in the interaction domain with that in the social domain, and computes the BPR loss for social domain embeddings, which allow similar users have similar preferences. However, it has several drawbacks. (1) It overlooks the noise in interaction domain, thus failing to ensure the authenticity of the learned user preferences. (2) BPR loss for social network link prediction based on noisy social network actually defeats the purpose of preference-guided denoising for social domain due to the information propogation between noisy social relations and interferes with the process of adaptive alignment of user representations in interaction and social domains.

Similarly, GDMSR computes the confidence of social relations and performs link prediction task for self-correction by utilizing the item embeddings in the interaction domain to represent user social embeddings. Additionally, it adaptively denoises the structure of the social network at regular intervals during the training process, achieving a certain level of noise reduction. However, it still overlooks the noise within the interaction domain, which compromises the overall denoising effect and the performance of the recommendation task.

In contrast, DCDSR not only utilizes the knowledge from interaction domain to guide the social network denoising, but also utilizes the social network to enhance the knowledge of interaction domain for denoising in turn and repeat the above process for adaptive optimizing, which realizes the bidirectional supervision of interaction and social domains. Meanwhile, the GNN-encoded user social embeddings are only served for the contrastive learning task and will not interfere with the recommendation task.

In terms of comparison of running time, we have recorded the time cost per epoch during the training process of each model. It can be observed from Fig. 5 and Table VI that across the three datasets, the per-epoch time cost of DCDSR is moderate compared with all other models. On the one hand, compared with the GSR models including LightGCN, NGCF, and DiffNet, DCDSR not only includes basic GNN modules but also integrates a contrastive learning module. Therefore, it is reasonable that DCDSR has a longer per-epoch runtime compared with them. On the other hand, compared with other models that introduce contrastive learning or knowledge distillation module, DCDSR demonstrates a significant advantage in terms of per-epoch time cost. Furthermore, from the perspective of the overall training process, DCDSR converges in fewer than 50 epochs across the three datasets, significantly reducing the total training time. This is attributed to the embedding-space collaborative denoising module, which accelerates the model's convergence. Therefore, it can be concluded that DCDSR achieves outstanding performance with relatively low total running time.

F. Ablation Study (RQ2)

In this section, we evaluate the effectiveness of the major components of DCDSR and further analyze the dual-domain embedding collaborative perturbation and AC-InfoNCE.

1) Effectiveness of Major Components: To test the effectiveness of each important component of DCDSR, we performed ablation experiments on them. Specifically, we compare the performance between DCDSR and the following ablation variants of DCDSR: (1) "DCDSR-RD": removing denoising for interaction domain at both structure level and embedding space while only adopt preference-guided social domain denoising. (2) "DCDSR-SD": removing the structure-level collaborative denoising module used to produce cleaner social network and interaction graph. (3) "DCDSR-ED": removing the embedding-space collaborative denoising module which restores the noise cross-domain diffusion process to learn denoised user and item embeddings. The results are shown in Fig. 6.

TABLE VI
COMBINED COMPARISON OF RECOMMENDATION PERFORMANCE (RECALL@20) AND TRAINING EFFICIENCY (S/EPOCH) OF DCDSR WITH GDMSR

Datasets	Ciao		Yelp2		Douban	
Method	Recall@3	s/Epoch	Recall@20	s/Epoch	Recall@20	s/Epoch
DCDSR	0.1426	2	0.4517	13	0.4561	8
GDMSR	0.1138	13	0.3405	25	0.3496	30

TABLE VII
PERFORMANCE COMPARISON BETWEEN DIFFERENT EMBEDDING PERTURBATION METHODS

Datasets	Metrics	DCDSR_{RP}	DCDSR_{CP}	Improve.%
Douban-Book	R@20	0.1725	0.1794	+4.00%
	N@20	0.1664	0.1726	+3.73%
Yelp	R@20	0.1230	0.1346	+9.43%
	N@20	0.0746	0.0820	+9.92%
Douban-Movie	R@20	0.1053	0.1116	+5.98%
	N@20	0.2553	0.2663	+4.31%

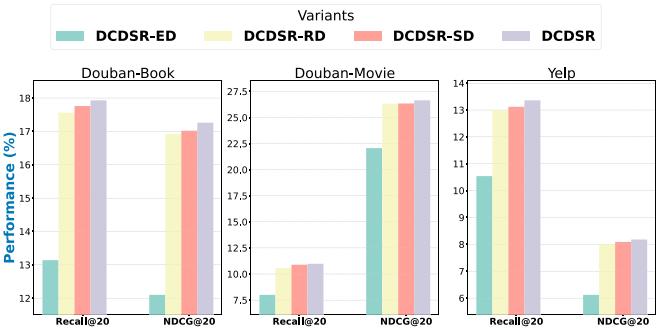

Fig. 6. Contributions of each major component in DCDSR.

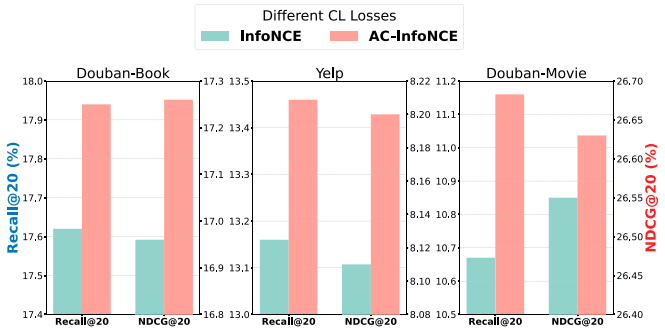

Fig. 7. Performance comparison of DCDSR with InfoNCE and AC-InfoNCE.

By comparison, we find that DCDSR performs better than other all ablation variants. Moreover, by comparing each variant we can draw the following conclusions. (1) The performance of DCDSR-RD shows a significant decline compared with DCDSR, which demonstrates that denoising for social domain is severely affected when there exists noise in interaction domain. This proves the correctness of our motivation and necessity of our work. (2) DCDSR-SD exhibits a notable performance decrease compared with DCDSR, indicating that ignoring structure-level collaborative denoising and employing noisy social network and interaction graph for GNN encoding allows the recommendation task to be greatly disturbed by structural noise. (3) DCDSR-ED experiences a substantial performance drop compared with DCDSR, suggesting that the embedding-space collaborative denoising module plays a crucial role in the model. It mitigates the adverse effects of noise cross-domain diffusion. In other words, It can discriminate and cut down the noise that escape during structure-level collaborative denoising.

2) Embedding Collaborative Perturbation Versus Embedding Random Perturbation: Embedding random perturbation (RP) is a common method of data augmentation for GCL, where perturbed embeddings are obtained by adding noise sample from a certain distribution such as Gaussian distribution [48] to the original embeddings. However, we argue that employing the random noise used for embedding perturbation does not have a specific semantic meaning and ignores the source of the noise, making the data augmentation imprecise. In contrast, the dual-domain embedding collaborative perturbation (CP) proposed by DCDSR specifies the source and semantics of the noise and imaginatively restores the noise cross-domain diffusion generated in structure-level collaborative denoising module, thus improving the explainability of data augmentation, which is conducive to learning more accurate and robust embeddings. As shown in Table VII, DCDSR_{CP} performs more superiorly than DCDSR_{RP} on all three datasets, thus demonstrating the effectiveness of the embedding collaborative perturbation.

3) AC-InfoNCE Versus InfoNCE: To verify the superiority of our proposed AC-InfoNCE, we compare the performance of DCDSR using AC-InfoNCE with that of DCDSR using InfoNCE. As shown in Fig. 7, on three datasets, replacing InfoNCE with AC-InfoNCE brings significant performance gains to DCDSR, confirming that AC-InfoNCE does allievate the gradient bias and data sparsity issues. The consistent improvement across different datasets demonstrates that AC-InfoNCE is a more powerful contrastive learning loss function and can be extended to other research fields.

G. Noise Resistance Evaluation (RQ3)

To evaluate the noise resistance of DCDSR, we conduct noise attack experiments on DCDSR, DSL, SEPT, and S^2-MHCN. We randomly generated fabricated user–item interactions with different ratios (i.e., 0.1, 0.2, and 0.3) to reconstruct the interaction data, and the variation of model performance with different noise ratios is shown in the following Fig. 8. It can be seen that on both douban-book and yelp, the performance of DCDSR with different ratios of noise attack is less degraded than other models. We attribute this to the following reasons. 1) The structure-level collaborative denoising module introduces denoising for interaction graph, which uses high-quality social information to play a supervised role in reconstructing a relatively clean interaction graph. This provides the model with a defensive capability in the face of interaction noise. 2)

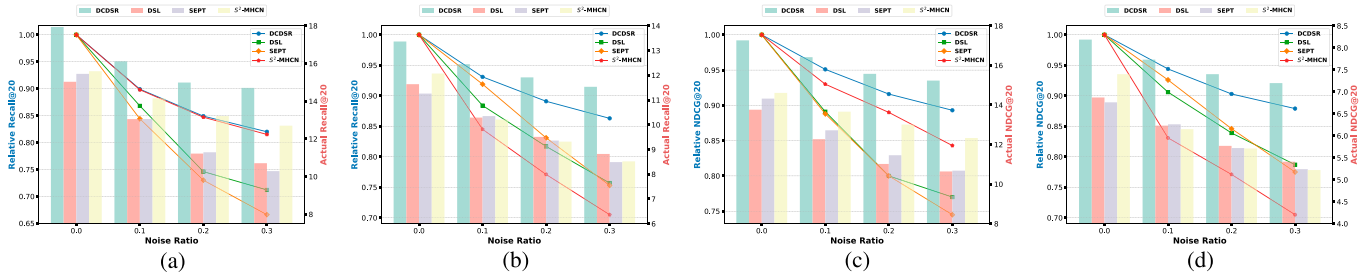

Fig. 8. Relative Recommendation Recall@20 and NDCG@20 w.r.t. different noise ratio on Douban-Book and Yelp datasets. (a) Douban-book. (b) Yelp. (c) Douban-book. (d) Yelp.

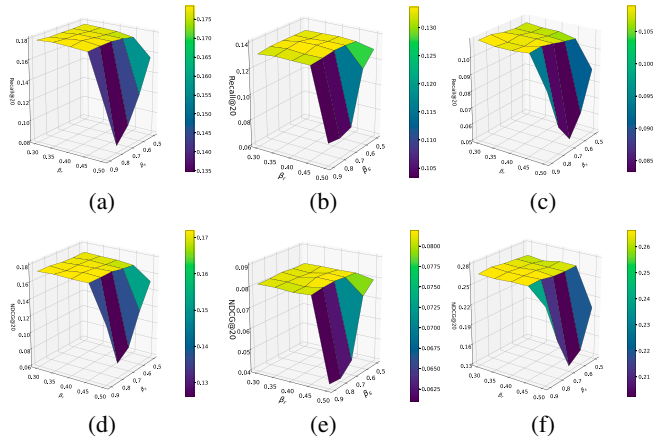

Fig. 9. Recommendation Recall@20 and NDCG@20 w.r.t. β_s and β_r on different datasets. (a) Douban-book. (b) Yelp. (c) Douban-movie. (d) Douban-book. (e) Yelp. (f) Douban-movie

The embedding-space collaborative denoising module is based on contrastive learning, which allows interaction domain to effectively absorb the auxiliary information from the social domain, thus providing more self-supervised signals for the recommendation task, and enabling the model to have the ability to automatically identify noise.

H. Parameter Sensitivity Analysis (RQ4)

In this section, we explore the impact of some key hyperparameters on model performance.

1) Impacts of β_s and β_r: In the structure-level collaborative denoising module, the preference consistency threshold β_s and the interaction compatibility threshold β_r are crucial. They directly influence the retention or deletion of edges within the social network and interaction graph. To assess their impacts, we conduct experiments on three datasets, holding other parameters constant while varying β_s and β_r within the ranges [0.5, 0.6, 0.7, 0.8, 0.9] and [0.3, 0.35, 0.4, 0.45, 0.5], respectively. Fig. 9 illustrates the effects of different β_s and β_r settings on Recall@20 and NDCG@20 metrics.

First, we analyze their impacts on model performance separately. For β_s, a clear trend is observed across all three datasets: as β_s increases, DCDSR performance also improves, peaking at optimal values (0.8 for Douban-Book, 0.7 for Yelp, and 0.8 for Douban-Movie) before declining with further increases. This observation indicates that higher values of β_s, which maintain social relations with greater preference consistency, improve the quality of social network and thereby facilitate the denoising for interaction graph. Additionally, this leads to more reliable user social embeddings learned via GNN, which benefits collaborative denoising in the embedding space. The model performance associated with β_r initially improves as the threshold increases, achieving an optimal value at 0.4 for Douban-Book, Yelp, and Douban-Movie. Beyond this peak, performance begins to decline. We believe that the deletion of edges for the original interaction graph needs to be careful, because a large β_r will cause too many interaction edges to be deleted thus aggravating the data sparsity problem, while too small β_r will result in noisy interaction data escaping filtered.

Subsequently, from a synergistic perspective, we conclude that DCDSR recursively repeats such pattern: ensuring that there is a large preference consistency (i.e., β_s) among social neighbors to get a social network that contains high-quality social information. Based on this, fusing social information to enhance the user embeddings in interaction domain, and then removes interaction edges with low user-item compatibilty to refine interaction graph while ensuring that the supervisory signals are not drastically reduced.

2) Impacts of $\lambda 1$, $\lambda 2$, and λ_3: In the embedding-space collaborative denoising module, three contrastive learning losses ($\mathcal{L}_{cl}^r$, $\mathcal{L}_{cl}^s$, $\mathcal{L}_{cl}^i$) guide the robust embedding learning process for the user/item. Assigning different weights to these three losses plays a crucial role in model training. Therefore, on all three datasets, we fixed the other parameters and adjusted them in the range of [0, 0.1, 0.2, 0.3, 0.4] to observe how the model performance changes. Observing Fig. 10, on Yelp and Douban-Movie, for three loss weights, the model obtains the best performance when they are all taken to 0.1. On Douban-Book, the model achieves its best performance when $\lambda 1$, $\lambda 2$, and λ_3 are taken to 0.2, 0.3, and 0.1. This suggests that Douban-Book dataset may contain more noisy interactions and social relations and thus requires a higher degree of denoising. Overall, for each weight, on all three datasets, the model performance increases first and then decreases as it increases, and the model performs poorly when it takes zero. This suggests that each contrastive learning loss is effective for embedding space denoising, while it should not be over-represented to prevent it from inhibiting the optimization of recommendation task.

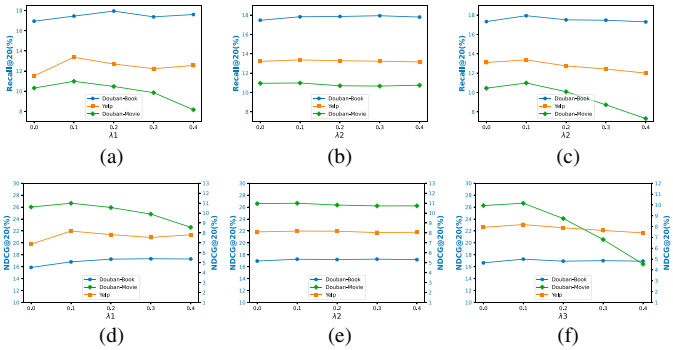

Fig. 10. Recommendation Recall@20 and NDCG@20 w.r.t. different weights of each contrastive loss on Douban-Book, Yelp, and Douban-Movie datasets. (a) $\lambda 1$. (b) $\lambda 2$. (c) $\lambda 3$. (d) $\lambda 1$. (e) $\lambda 2$. (f) $\lambda 3$.

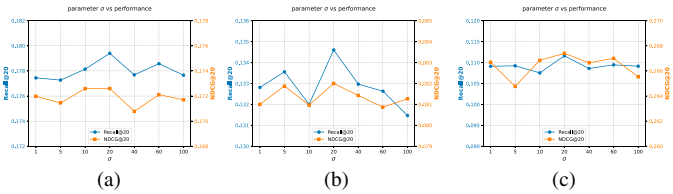

Fig. 11. Recommendation Recall@20 and NDCG@20 w.r.t. σ on different datasets. (a) Douban-book. (b) Yelp. (c) Douban-movie.

3) Impacts of σ: In the Gaussian kernel similarity, σ controls the sensitivity of the kernel function to the Euclidean distance. Specifically, a smaller σ makes the kernel function more sensitive to distance, where only very close embeddings are considered similar. A larger σ makes the kernel function smoother, allowing even distant embeddings to be considered somewhat similar. Meanwhile, the value of σ affects the distribution of Gaussian kernel similarity, thereby influencing the importance of Gaussian kernel similarity in preference consistency.

Therefore, we intend to explore a proper value of σ. This ensures that the model captures both the directional similarity and the spatial distribution of embeddings, improving the robustness and accuracy of the similarity calculation. Specifically, we conduct a grid search within a range that spans multiple magnitudes [1, 5, 10, 20, 40, 60, 100].

The experimental results are presented in Fig. 11. We can observe that across the three datasets, the model achieves the best performance when σ is set to 20. Therefore, we considered 20 to be an appropriate choice for σ.

4) Impacts of τ: The role of the temperature parameter τ in AC-InfoNCE is to control the embedding distribution of positive and negative samples. A smaller τ makes the model focus more on similar samples (including easy positive and hard negative samples), while a larger τ ensures attention to a broader range of samples, leading to a more uniform distribution of positive and negative samples. However, when the τ is too small, the model tends to focus only on high-similarity positive samples, resulting in overfitting of model's training. On the other hand, an excessively large τ makes it

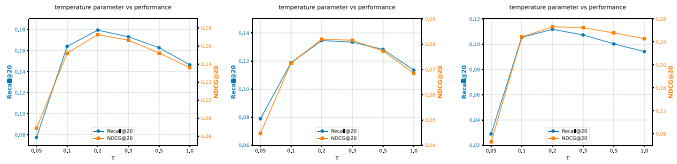

Fig. 12. Recommendation Recall@20 and NDCG@20 w.r.t. τ on different datasets. (a) Douban-book. (b) Yelp. (c) Douban-movie.

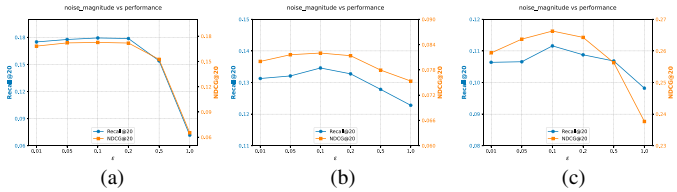

Fig. 13. Recommendation Recall@20 and NDCG@20 w.r.t. ϵ on different datasets. (a) Douban-book. (b) Yelp. (c) Douban-movie.

difficult to distinguish between positive and negative samples. Therefore, selecting an appropriate value of τ is crucial for effective embedding learning. Under the condition that other hyperparameters remain unchanged, we adjust the temperature parameter τ within the range [0.05, 0.1, 0.2, 0.3, 0.5, 1.0] and observe the changes in model performance. It can be observed from Fig. 12, across the three datasets, as the τ increases, the model's performance initially improves and then declines. The performance peaks when the τ is set to 0.2, indicating that 0.2 is the optimal value for the temperature parameter.

5) Impacts of ϵ: In collaborative embedding perturbation, the magnitude of perturbation noise ϵ plays a crucial role in learning denoised embeddings. The generated perturbation noise carries specific domain information, and an appropriate noise magnitude can effectively mitigate cross-domain diffusion noise in the original embeddings. Therefore, we empirically adjust the noise magnitude ϵ within the range of [0.01, 0.05, 0.1, 0.2, 0.5, 1.0] while keeping other hyperparameters unchanged, and observe the changes in model performance. As Fig. 13 shows, across all three datasets, the model performance first improves and then declines as the noise magnitude increases. The model achieves the best performance when the ϵ is set to 0.1. Therefore, we set the magnitude of perturbation noise to 0.1.

V. Conclusion and Future Work

In this article, we explore the noise issues in social recommendation and propose a novel denoising social recommendation model, called DCDSR. DCDSR contains two denoising modules, namely, structure-level collaborative denoising and embedding-space collaborative denoising. The structure-level collaborative denoising process is applied to construct cleaner social network and interaction graph with a collaborative denoising paradigm. The embedding-space collaborative denoising module simulates the raw embeddings unaffected by noise cross-domain diffusion with dual-domain

embedding collaborative perturbation and learns denoised embeddings with contrastive learning. Ultimately, DCDSR learns high-quality and robust user/item embeddings for recommendation through multi-task joint training. Experimental results on various datasets demonstrate that DCDSR outperforms the state-of-the-art models including graph-based recommendation, graph-based social recommendation and denoising social recommendation, validating the superiority of DCDSR in dealing with the noise issues. More importantly, DCDSR not only solves the noise issue in social network, but also collaboratively reduces the noise of interaction data, which truly plays the auxiliary role of social network for collaborative filtering.

In the future, we will conduct a more granular investigation into the noise issues in social recommendation to better perform denoising task in social recommendation. Furthermore, both noise identification methods and noise filtering techniques are worthwhile directions for research.

REFERENCES

[1] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. ACM Conf. Recommender Syst.*, 2007, pp. 17–24.

[2] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proc. 14th Int. Conf. Knowl. Discovery Data Mining (ACM SIGKDD)*, 2008, pp. 426–434.

[3] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[4] S. Chen, S. Owusu, and L. Zhou, "Social network based recommendation systems: A short survey," in *Proc. Int. Conf. Social Comput.*, Piscataway, NJ, USA: IEEE Press, 2013, pp. 882–885.

[5] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, 2001.

[6] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[7] Z. Zhang, M. Dong, K. Ota, and Y. Kudo, "Alleviating new user cold-start in user-based collaborative filtering via bipartite network," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 3, pp. 672–685, Jun. 2020.

[8] P. V. Marsden and N. E. Friedkin, "Network studies of social influence," *Sociolog. Methods Res.*, vol. 22, no. 1, pp. 127–151, 1993.

[9] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proc. 42nd Int. Conf. Res. Develop. Inf. Retrieval (ACM SIGIR)*, 2019, pp. 235–244.

[10] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, "DiffNet++: A neural influence and interest diffusion network for social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4753–4766, Oct. 2020.

[11] X. Cai, W. Guo, M. Zhao, Z. Cui, and J. Chen, "A knowledge graph-based many-objective model for explainable social recommendation," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 6, pp. 3021–3030, Dec. 2023.

[12] Z. Huang et al., "Social group recommendation with TrAdaBoost," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 5, pp. 1278–1287, Oct. 2020.

[13] T. Liang, L. Ma, W. Zhang, H. Xu, C. Xia, and Y. Yin, "Content-aware recommendation via dynamic heterogeneous graph convolutional network," *Knowl.-Based Syst.*, vol. 251, 2022, Art. no. 109185.

[14] L. Yang, Z. Liu, Y. Wang, C. Wang, Z. Fan, and P. S. Yu, "Large-scale personalized video game recommendation via social-aware contextualized graph neural network," in *Proc. ACM Web Conf.*, 2022, pp. 3376–3386.

[15] X. Sha, Z. Sun, and J. Zhang, "Disentangling multi-facet social relations for recommendation," *IEEE Trans. Comput. Social Syst.*, vol. 9, no. 3, pp. 867–878, Jun. 2022.

[16] L. Sang, Y. Wang, Y. Zhang, and X. Wu, "Denoising heterogeneous graph pre-training framework for recommendation," *ACM Trans. Inf. Syst.*, 2024.

[17] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proc. Web Conf.*, 2021, pp. 413–424.

[18] J. Yu, H. Yin, M. Gao, X. Xia, X. Zhang, and N. Q. Viet Hung, "Socially-aware self-supervised tri-training for recommendation," in *Proc. 27th ACM Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2021, pp. 2084–2092.

[19] J. Wu, W. Fan, J. Chen, S. Liu, Q. Li, and K. Tang, "Disentangled contrastive learning for social recommendation," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 4570–4574.

[20] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," *Acad. Manage. Rev.*, vol. 20, no. 3, pp. 709–734, 1995.

[21] Y. Tao, Y. Li, S. Zhang, Z. Hou, and Z. Wu, "Revisiting graph based social recommendation: A distillation enhanced social graph network," in *Proc. ACM Web Conf.*, 2022, pp. 2830–2838.

[22] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.

[23] W. Wang, F. Feng, X. He, L. Nie, and T.-S. Chua, "Denoising implicit feedback for recommendation," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 373–381.

[24] Y. Zhang, Y. Zhang, D. Yan, S. Deng, and Y. Yang, "Revisiting graph-based recommender systems from the perspective of variational auto-encoder," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–28, 2023.

[25] F. Wang, H. Zhu, G. Srivastava, S. Li, M. R. Khosravi, and L. Qi, "Robust collaborative filtering recommendation with user-item-trust records," *IEEE Trans. Comput. Social Syst.*, vol. 9, no. 4, pp. 986–996, Aug. 2022.

[26] T. Wang, L. Xia, and C. Huang, "Denoised self-augmented learning for social recommendation," 2023, *arXiv:2305.12685*.

[27] Y. Quan, J. Ding, C. Gao, L. Yi, D. Jin, and Y. Li, "Robust preference-guided denoising for graph based social recommendation," in *Proc. ACM Web Conf.*, 2023, pp. 1097–1108.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[29] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

[30] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[31] S. Xiao, S. Wang, Y. Dai, and W. Guo, "Graph neural networks in node classification: survey and evaluation," *Mach. Vis. Appl.*, vol. 33, no. 1, p. 4, 2022.

[32] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2019, pp. 165–174.

[33] K.-H. N. Bui, J. Cho, and H. Yi, "Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues," *Appl. Intell.*, vol. 52, no. 3, pp. 2763–2774, 2022.

[34] A. Bastos et al., "RECON: relation extraction using knowledge graph context in a graph neural network," in *Proc. Web Conf.*, 2021, pp. 1673–1685.

[35] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2020, pp. 639–648.

[36] Y. Zhang, Y. Zhang, D. Yan, Q. He, and Y. Yang, "NIE-GCN: Neighbor item embedding-aware graph convolutional network for recommendation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 54, no. 5, pp. 2810–2821, May 2024.

[37] N. Li, C. Gao, D. Jin, and Q. Liao, "Disentangled modeling of social homophily and influence for social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5738–5751, Jun. 2023.

[38] J. Liao et al., "SocialLGN: Light graph convolution network for social recommendation," *Inf. Sci.*, vol. 589, pp. 595–607, Apr. 2022.

[39] Q. Wu et al., "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 2091–2102.

[40] L. Xia, Y. Shao, C. Huang, Y. Xu, H. Xu, and J. Pei, "Disentangled graph social recommendation," 2023, *arXiv:2303.07810*.

[41] L. Sang, Y. Wang, Y. Zhang, Y. Zhang, and X. Wu, "Intent-guided heterogeneous graph contrastive learning for recommendation," 2024, *arXiv:2407.17234*.

[42] W. Ma et al., "Madm: A model-agnostic denoising module for graph-based social recommendation," in *Proc. 17th ACM Int. Conf. Web Search Data Mining*, (ser. WSDM), New York, NY, USA: ACM, 2024, pp. 501–509. [Online]. Available: https://doi.org/10.1145/3616855.3635784
[43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2020, pp. 1597–1607.
[44] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," 2021, *arXiv:2104.08821*.
[45] J. Wu et al., "Self-supervised graph learning for recommendation," in *Proc. 44th Int. ACM Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2021, pp. 726–735.
[46] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, *JMLR Workshop Conf. Proc.*, 2010, pp. 297–304.
[47] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," 2018, *arXiv:1808.06670*.
[48] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? simple graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 1294–1303.
[49] J. Ji, B. Zhang, J. Yu, X. Zhang, D. Qiu, and B. Zhang, "Relationship-aware contrastive learning for social recommendations," *Inf. Sci.*, vol. 629, pp. 778–797, Jun. 2023.
[50] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–20, Oct. 2009.
[51] H. Ye, X. Li, Y. Yao, and H. Tong, "Towards robust neural graph collaborative filtering via structure denoising and embedding perturbation," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–28, 2023.
[52] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 566–577.
[53] P. K. Diederik, "Adam: A method for stochastic optimization," 2014.
[54] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, *JMLR Workshop Conf. Proc.*, 2010, pp. 249–256.

Wenjie Chen received the bachelor's degree in computer science and technology from Anhui Normal University, Wuhu, China, in 2022. He is currently working toward the master's degree in computer science and technology with Anhui University, Hefei, China.

His research interests include graph neural networks, recommender systems, and data mining.

Yi Zhang received the bachelor's degree in 2020 from Anhui University, Hefei, China, where he is currently working toward the Ph.D. degree, all in computer science and technology.

He has publications in several top conferences and journals, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS,IEEE TRANSACTIONS ON BIG DATA, *ACM Transactions on Information Systems*, and *ACM SIGIR Special Interest Group on Information Retrieval*, etc. His research interests include graph neural network, personalized recommender systems, and service computing.

Honghao Li received the bachelor's degree in software engineering from Bengbu University, Bengbu, China, in 2022. He is currently working toward the Ph.D. degree in computer science and technology at Anhui University, Hefei, China.

His research interests include graph neural networks, recommender systems, and data mining.

Lei Sang received the Ph.D. degree from University of Technology Sydney, Sydney, Australia, in 2021.

Currently, he is a Lecturer with the School of Computer Science and Technology, Anhui University, Anhui, China. His research interests include natural language processing, data mining, and recommendation systems.

Yiwen Zhang received the Ph.D. degree in management science and engineering from Hefei University of Technology, Hefei, China, in 2013.

Currently, he is a Full Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. He has published more than 70 papers in highly regarded conferences and journals, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, *ACM Transactions on Information Systems*, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *ACM Transactions on Knowledge Discovery from Data*, *Special Interest Group on Information Retrieval*, *International Conference on Service-Oriented Computing*, *International Conference on Web Services*, etc. His research interests include service computing, cloud computing, and big data analytics.