

# Balancing User Preferences by Social Networks: A Condition-Guided Social Recommendation Model for Mitigating Popularity Bias

Xin He<sup>a</sup>, Wenqi Fan<sup>b</sup>, Ruobing Wang<sup>a</sup>, Yili Wang<sup>a</sup>, Ying Wang<sup>c</sup>, Shirui Pan<sup>d</sup>, Xin Wang<sup>a</sup>

<sup>a</sup> *The Department of Computer Science and Technology, the School of Artificial Intelligence (SAI), at Jilin University, 2699 Qianjin Street, Chang Chun, 130000, Jilin, China*

<sup>b</sup> *The Hong Kong Polytechnic University, Kowloon, Hong Kong, 000000, Hong Kong, China*

<sup>c</sup> *The Department of Computer Science and Technology, College of Computer Science and Technology, at Jilin University, 2699 Qianjin Street, Chang Chun, 130000, Jilin, China*

<sup>d</sup> *Griffith University, Parklands Drive, Southport, 4222, Queensland, Australia*

---

## Abstract

Social recommendation models weave social interactions into their design to provide uniquely personalized recommendation results for users. However, social networks not only amplify the popularity bias in recommendation models, resulting in more frequent recommendation of hot items and fewer long-tail items, but also include a substantial amount of redundant information that is essentially meaningless for the model's performance. Existing social recommendation models often integrate the entire social network directly, with little effort to filter or adjust social information to mitigate popularity bias introduced by the social network. In this paper, we propose a Condition-Guided Social Recommendation Model (named CGSoRec) to mitigate the model's popularity bias by denoising the social network and adjusting the weights of user's social preferences. More specifically, CGSoRec first includes a Condition-Guided Social Denoising Model (CSD) to remove redundant social relations in the social network for capturing users' social preferences with items more precisely. Then, CGSoRec calculates users' social preferences based on denoised social network and adjusts the weights in users' social preferences to make them can counteract the popularity bias present in the recommendation model. At last, CGSoRec includes a Condition-Guided Diffusion Recommendation Model (CGD) to introduce the adjusted social

preferences as conditions to control the recommendation results for a de-biased direction. Comprehensive experiments on three real-world datasets demonstrate the effectiveness of our proposed method. The anonymous code is in: <https://github.com/hexin5515/CGSoRec>.

*Keywords:* Recommendation System, Social Recommendation, Popularity Bias, Conditional Diffusion Model

## 1. Introduction

Over the past decade, social recommendation has become a vital component of recommendation systems. Recent studies [1, 2, 3, 4, 5] have demonstrated that incorporating user-user social relationships can enhance models' ability to effectively capture collaborative signals between users and items in certain scenarios. This improvement is based on the theory of social homogeneity [6, 7], which posits that users with a social relationship are likely to share similar preferences. Social homogeneity has been corroborated in a diverse range of application domains within recommendation systems [8, 9, 10].

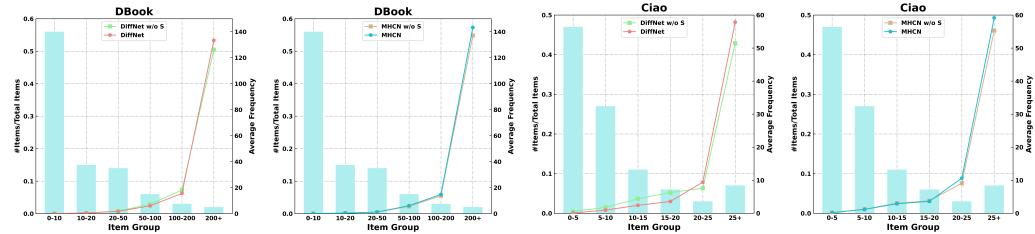


Figure 1: Popularity bias comparison between introduced and removed Social Network versions of DiffNet and MHCN model.

Despite their superior effectiveness, a key limitation of social recommendation models is that they tend to amplify popularity bias [11] which makes popular items receive more recommendation. Fig. 1 presents that the social recommendation methods (Diffnet [12] and MHCN [1]) lead to amplifying popularity bias on a real-world Ciao and DBBook dataset [13]. We train two classical social recommendation models and quantify item frequencies in the top- $K$  recommendation across all users [11]. As can be seen, compared to the versions DiffNet (MHCN) w/o S that removes social information, DiffNet (MHCN) increases the frequency of recommendation for hot items. This exacerbation may enhance the performance of social recommendation models

on traditional recommendation datasets, but in scenarios requiring fair recommendation [11, 14], it often leads to a decrease in model performance. Nevertheless, there is still no research focusing on specifically designing models to mitigate popularity bias in social recommendation. This motivates us to pose the following research question: **how to design a method to mitigate the popularity bias due to the introduction of social relationships?**

To answer this question, we first examine why social networks amplify the popularity bias in social recommendation models. We quantify the distribution of items favored by a user’s social neighbors, which named the user’s social preference [15]. As shown in Fig. 2, in a users’ social preferences, hot items are recommended more often than long-tail items, exhibiting a trend similar to the model’s inherent popularity bias shown in Fig. 1. Therefore, the integration of social networks into the recommendation model further amplifies its popularity bias, leading to an increased recommendation of hot items by the model. It is important to note that users’ social preferences for long-tail items are also present but less pronounced compared to their preferences for hot items. Therefore, inspired by knowledge editing method [16, 17, 18] in the field of Computer Vision (CV) and Natural Language Processing (NLP), we propose to adjust the weights in user social preferences. This adjustment aims to reveal preferences that counteract the inherent bias within the recommendation model, thereby mitigating its popularity bias.

Recently, with the development of diffusion models in the field of recommendation systems, some studies [19, 20, 21, 22] have shown that diffusion recommendation models can model the complex interactions between users and items more accurately compared to traditional matrix factorization recommendation models and graph neural network recommendation models. Besides, with the development of Condition-Guided Diffusion Models in the fields of CV and NLP, numerous studies [23, 24, 25, 26, 27] have demonstrated that Condition-Guided Diffusion Models can proficiently modulate generation results as desired. This motivates our investigation into the generation of unbiased recommendation results through the utilization of a condition-guided diffusion recommendation model. However, this investigation also presents us with

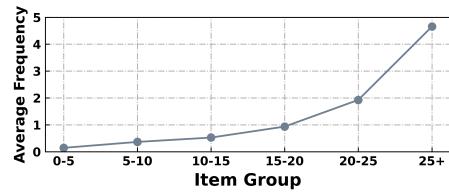


Figure 2: Average frequency of items recommended based on a user’s social preferences.

two distinct challenges: (1) The incorporation of social information often exacerbates the popularity bias inherent in recommendation models. Therefore, determining how to adjust users' social preferences when incorporating social information as a condition into the recommendation model is crucial for mitigating the popularity bias. (2) Nearly all social networks contain an abundance of redundant relations [2, 28, 29], which are not only irrelevant for improving the performance of the social recommendation models but also detrimentally affect their performance. Therefore, it is critical to remove these redundant social relations in social networks when taking advantage of social information.

To address the above issues, in this paper, we propose a Condition-Guided Social Recommendation Model (named **CGSoRec**), which can mitigate the popularity bias by conditionally steering the recommendation results in alignment with the user's social preferences. More specifically, we first propose a Condition-Guided Social Denoising Model (CSD) to remove redundant or noisy social relations in social networks to reflect users' social preferences more accurately. Then, we derive users' social preferences based on denoised social networks and adjust the weights within these preferences to counteract the inherent popularity bias in the model. At last, we propose a Condition-Guided Diffusion Recommendation Model (CGD) to introduce users' social preferences into diffusion recommendation models as conditions and mitigate the popularity bias by adjusting the weights of hot and long-tail items within user's social preferences.

In summary, this work makes the following contributions:

- We are the first to address the issue of popularity bias in social recommendation models by incorporating adjusted social preferences as conditions in a Condition-Guided Diffusion Recommendation Model (CGD).
- We innovatively propose a Condition-Guided Social Denoising Model (CSD) to eliminate redundant information in social networks, thereby enabling the model to capture users' social preferences more accurately.
- We conduct comprehensive experiments and ablation studies on three real-world datasets to demonstrate the effectiveness of our proposed method.

## 2. Related Work

### 2.1. Social Recommendation Model

With the development of social media, leveraging social relations for recommendation has garnered considerable interest in the field in recent years [30, 12, 31, 32, 33]. More specifically, existing social recommendation models leverages various techniques [34, 35, 36] to integrate social networks into their framework, aiming to enhance the overall performance of these models. For instance, GraphRec[31] utilizes the capabilities of graph neural networks to improve the representation learning of users and items within the context of social recommendation. DiffNet [12] employs a GNNs model to simulate the process of social influence propagation, thereby achieving more refined representations of users and items. MHCN [37] integrates self-supervised learning and hypergraph convolution techniques to augment relational data, focusing on encoding higher-order and complex connection patterns, thereby significantly advancing the efficacy of social recommendation models.

### 2.2. Generative Recommendation Model

Over the past decade, there has been a rising fascination with generative models such as Variational Auto-Encoders (VAEs) [38, 39, 40] and Denoising Diffusion Probabilistic Model (DDPM) [41, 42, 43] for personalized recommendation due to their remarkable capacity of modeling non-linear user-item interactions and denoising. For example, Mult-VAE [44] proposes a variant of vae for collaborative filtering on implicit feedback data, which enable to go beyond linear factor models with limited modeling capacity. DiffRec [19] proposes a new recommendation paradigm for generative recommendation models and improves the performance of the model through a time-aware reweighting strategy.

### 2.3. Condition-Guided Diffusion Model

Meanwhile, with the development of conditional diffusion model, numerous studies [26, 24, 45, 46] have demonstrated the advantages of employing conditional diffusion models to improve the quality of the generated results. For example, the classifier-guidance diffusion model [24] employs a conditional classifier to direct the diffusion process, facilitating the creation of outputs that are more precise and aligned with specified conditions. The classifier-free diffusion model [26] incorporates conditional guidance to steer

the generation process, maintaining its training procedure unchanged while allowing for the creation of diverse results guided by specific conditions. In this work, we explore how to denoise social networks and utilize the adjusted user social preference to control the recommendation results of the diffusion recommendation model for mitigating the popularity bias.

### 3. Preliminaries

In this section, we introduce definitions and notations that are used throughout the paper. Let  $U = \{u_1, u_2, \dots, u_m\}$  and  $V = \{v_1, v_2, \dots, v_n\}$  be the sets of users and items, respectively, where  $m$  is the number of users and  $n$  is the number of items. We denote  $\mathbf{R} \in \mathbb{R}^{m \times n}$  as a user-item interaction matrix, where  $r_{ij} = 1$  if user  $u_i$  has interacted with item  $v_j$ , otherwise  $r_{ij} = 0$ . We denote  $\mathbf{S} \in \mathbb{R}^{m \times m}$  as a user-user social matrix, where  $s_{ii'} = 1$  if user  $u_i$  has a social relation with user  $u_{i'}$ , otherwise  $s_{ii'} = 0$ . In this work, since the data in the social network cannot be injected directly into the user's interaction vector  $x_i$ , we utilize the user social preference matrix  $\mathbf{R}^s \in \mathbb{R}^{m \times n} = \mathbf{SR}$  to enhance the performance of the recommendation model. The row vector  $\mathbf{x}_i^s = [x_{i1}^s, x_{i2}^s, \dots, x_{in}^s]$  in the matrix  $\mathbf{R}^s$  denotes the interaction of the target user's social neighborhood with all items. Given  $\mathbf{R}$  and  $\mathbf{R}^s$ , our goal is to direct the model's denoising of interaction matrix  $\mathbf{R}$  using the social preference matrix  $\mathbf{R}^s$  for predicting users' unknown preferences towards items and mitigate the popularity bias.

### 4. The Proposed Method

In this section, we first introduce our proposed Condition-Guided Social Recommendation Model (**CGSoRec**). Then, we provide a comprehensive overview of the model training process and inferencing process.

#### 4.1. An Overview of the Proposed Method

The overall architecture of the proposed model is shown in Fig. 3, which consists of two main submodels: (1) **Condition-Guided Social Denoising Model (CSD)**, which is introduced to reflect users' social preference more accurately by introducing user-user interaction for co-purchasing long-tail items as conditions and removing redundant or noisy social relationships in social networks; (2) **Condition-Guided Diffusion Recommendation**

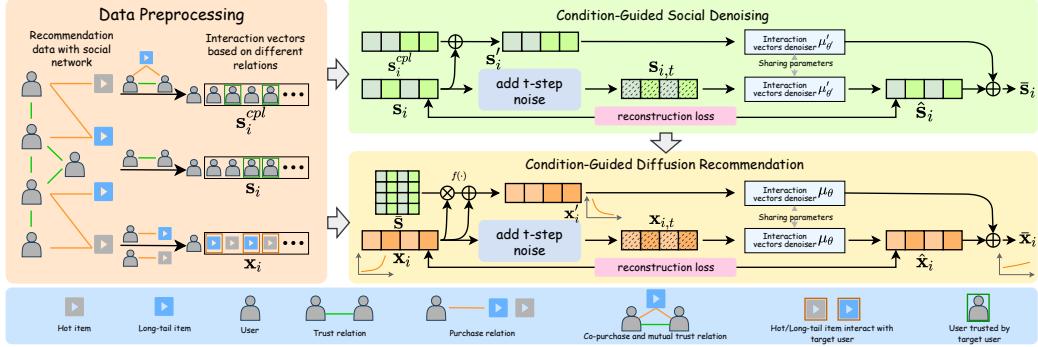


Figure 3: The overall framework of the proposed method.

**Model (CGD)**, which aims to mitigate the popularity bias of the recommendation model by incorporating the social information as a condition into the diffusion recommendation model and adjusting the weights of hot/long-tail items in the user’s social preference. Specifically, we first derive the user-item interaction vector  $\mathbf{x}_i$ , as well as the user-user interaction matrices  $\mathbf{s}_i^{cpl}$  and  $\mathbf{s}_i$ , based on varying user-item and user-user relationships. Then, we compute the interaction vector  $\mathbf{s}'_i$  to direct the CSD in the denoising process of the social interaction vector  $\mathbf{s}_i$ . At last, utilizing the denoised user social interaction vector  $\bar{\mathbf{s}}_i$ , we calculate the user’s social preferences and apply these preferences, adjusted for weight as  $\mathbf{x}'_i$ , to steer the CGD, thereby achieving bias-reduced recommendation outcomes  $\bar{\mathbf{x}}_i$ . Next, we will first introduce the general paradigm of the diffusion recommendation model, followed by a detailed description of the two submodels.

#### 4.2. Diffusion Recommendation Model

As one of the most representative diffusion recommendation methods, DiffRec [19] effectively fills in the missing values in user-item interaction vectors in a denoising manner, thereby providing personalized recommendations for users. Therefore, we adopt DiffRec as our target encoder for denoising user-item interaction vectors in user-item interactions matrix  $\mathbf{R} \in \mathbb{R}^{m \times n}$ . The forward diffusion process of the diffusion recommendation model can be formulated as follows:

$$q(\mathbf{x}_{i,t} | \mathbf{x}_{i,t-1}) = \mathcal{N}(\mathbf{x}_{i,t}; \sqrt{1 - \beta_t} \mathbf{x}_{i,t-1}, \beta_t \mathbf{I}), \quad (1)$$

where  $\beta_t$  controls the strength of the Gaussian noise added at each step of the forward diffusion process. We can directly obtain  $\mathbf{x}_{i,t}$  from  $\mathbf{x}_{i,0}$  due to

the additivity of two independent Gaussian noises [47] as:

$$q(\mathbf{x}_{i,t}|\mathbf{x}_{i,0}) = \mathcal{N}(\mathbf{x}_{i,t}; \sqrt{\bar{\alpha}_t}\mathbf{x}_{i,0}, (1 - \bar{\alpha}_t)\mathbf{I}), \quad (2)$$

where  $\alpha_t = 1 - \beta_t$ ,  $\bar{\alpha}_t = \prod_{t'=1}^t \alpha_{t'}$  and we can directly obtain  $\mathbf{x}_{i,t}$  as:  $\mathbf{x}_{i,t} = \sqrt{\bar{\alpha}_t}\mathbf{x}_{i,0} + \sqrt{1 - \bar{\alpha}_t}\epsilon$  with  $\epsilon \sim \mathcal{N}(0, 1)$ . Then, the diffusion recommendation model gradually recovers users' interactions by the denoising transition step:

$$p_\theta(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t}) = \mathcal{N}(\mathbf{x}_{i,t-1}; \mu_\theta(\mathbf{x}_{i,t}, t), \Sigma_\theta(\mathbf{x}_{i,t}, t)), \quad (3)$$

where  $\mu_\theta(\mathbf{x}_{i,t}, t)$  and  $\Sigma_\theta(\mathbf{x}_{i,t}, t)$  are the Gaussian parameters outputted by any neural networks with learnable parameters  $\theta$ . The diffusion recommendation model is mainly optimized by regulating the  $p_\theta(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t})$  to approximate tractable ground-truth transition step  $q(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t}, \mathbf{x}_{i,0})$  which can be rewritten as [47]:

$$q(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t}, \mathbf{x}_{i,0}) \propto \mathcal{N}(\mathbf{x}_{i,t-1}; \tilde{\mu}(\mathbf{x}_{i,t}, \mathbf{x}_{i,0}, t), \sigma^2(t)\mathbf{I}), \quad (4)$$

$$\begin{cases} \tilde{\mu}(\mathbf{x}_{i,t}, \mathbf{x}_{i,0}, t) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\mathbf{x}_{i,t} + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t}\mathbf{x}_{i,0}, \\ \sigma^2(t) = \frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}, \end{cases} \quad (5)$$

where  $\tilde{\mu}(\mathbf{x}_{i,t}, \mathbf{x}_{i,0}, t)$  and  $\sigma^2(t)$  are the mean and covariance of  $q(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t}, \mathbf{x}_{i,0})$  decoupled from Eq. (2). Naturally, the optimization objective of the diffusion recommendation model can be changed from forcing  $p_\theta(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t})$  to  $q(\mathbf{x}_{i,t-1}|\mathbf{x}_{i,t}, \mathbf{x}_{i,0})$  into pushing  $\mu_\theta(\mathbf{x}_{i,t}, t)$  be closed to  $\tilde{\mu}(\mathbf{x}_{i,t}, \mathbf{x}_{i,0}, t)$ . And  $\mu_\theta(\mathbf{x}_{i,t}, t)$  can be formulated as:

$$\mu_\theta(\mathbf{x}_{i,t}, t) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\mathbf{x}_{i,t} + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t}\hat{x}_\theta(\mathbf{x}_{i,t}, t), \quad (6)$$

where  $\hat{x}_\theta(\mathbf{x}_{i,t}, t)$  is the neural networks with learnable parameters  $\theta$ . With the user-item interaction vector  $\mathbf{x}_i$ , the goal of the diffusion recommendation model is to predict users' unknown preferences towards items by Eq. 3.

#### 4.3. Condition-Guided Social Denoising Model

Original social networks often contain excessive redundant information, which may lead to the model being unable to accurately reflect users' social preferences. Therefore, we propose a Condition-Guided Social Denoising Model (CSD) that removes redundant or noisy social relationships in social

networks. Similar to the diffusion recommendation model, we consider the row vector  $\mathbf{s}_i = [s_{i1}, s_{i2}, \dots, s_{im}]$  in the social matrix  $\mathbf{S} \in \mathbb{R}^{m \times m}$ , which encapsulates the user's social interaction records, as the input to CSD, and adds Gaussian noise to  $\mathbf{s}_i$  using a similar method as Diffusion Recommendation Model. Subsequently, a neural network  $\mu_{\theta'}$  is employed to learn the denoising process, which can be formulated as follows:

$$p_{\theta'}(\mathbf{s}_{i,t-1} | \mathbf{s}_{i,t}) = \mathcal{N}(\mathbf{s}_{i,t-1}; \mu_{\theta'}(\mathbf{s}_{i,t}, t), \Sigma_{\theta'}(\mathbf{s}_{i,t}, t)), \quad (7)$$

where  $\theta'$  is the learnable parameters of the Social Denoising Model. Then, we introduce co-purchase relationships as a condition in the CSD. Specifically, we utilize the user-item interaction matrix  $\mathbf{R}_l$  which records the user interactions with long-tail items, to derive the user-user interaction matrix  $\mathbf{S}^{cpl}$  for co-purchasing long-tail items. And we also incorporate the user-user interaction matrix  $\mathbf{S}^{cpl}$  for co-purchasing long-tail items into the user-user interaction matrix  $\mathbf{S}$  as condition so that CSD prioritizes users with similar long-tail item preferences:

$$\mathbf{S}^{cpl} = \mathbf{R}_l \mathbf{R}_l^T, \quad (8)$$

$$\mathbf{S}' = \delta \mathbf{J} \odot \mathbf{S}^{cpl} + \mathbf{S}, \quad (9)$$

where  $\mathbf{J} \in \mathbb{R}^{m \times n}$  is the matrix whose elements are all 1.  $s_{ij}^{cpl}$  denotes the sum of co-purchasing long-tail item relationships from user  $u_i$  to user  $u_j$ . The larger value of  $s_{ij}^{cpl}$  indicates that user  $u_i$  and  $u_j$  exhibit identical preferences for long-tail items.  $\odot$  denotes the element-wise product.  $\delta$  is a coefficient to control the effect of the condition on the model's denoising results. The denoising process can be formulated as follows:

$$p_{\theta'}(\mathbf{s}_{i,t-1} | \mathbf{s}_{i,t}, \mathbf{s}'_i) = \mathcal{N}(\mathbf{s}_{i,t-1}; \mu'_{\theta'}(\mathbf{s}_{i,t}, \mathbf{s}'_i, t), \Sigma_{\theta'}), \quad (10)$$

$$\mu'_{\theta'}(\mathbf{s}_{i,t}, \mathbf{s}'_i, t) = (1 - \eta)\mu_{\theta'}(\mathbf{s}_{i,t}, t) + \eta\mu_{\theta'}(\mathbf{s}'_i, t), \quad (11)$$

where  $\mathbf{s}'_i = [s'_{i1}, s'_{i2}, \dots, s'_{in}]$  is the row vectors in matrix  $\mathbf{S}'$  recording user-user social relationships and user co-purchase relationships for long-tail items.  $\mu'_{\theta'}(\mathbf{s}_{i,t}, \mathbf{s}'_i, t)$  and  $\Sigma_{\theta'} = \Sigma_{\theta'}(\mathbf{s}_{i,t}, t)$  are the Gaussian parameters by neural networks with learnable parameters  $\theta'$ . The hyper-parameter  $\eta$  is employed in our CSD to regulate the degree in the inferencing process integrating the

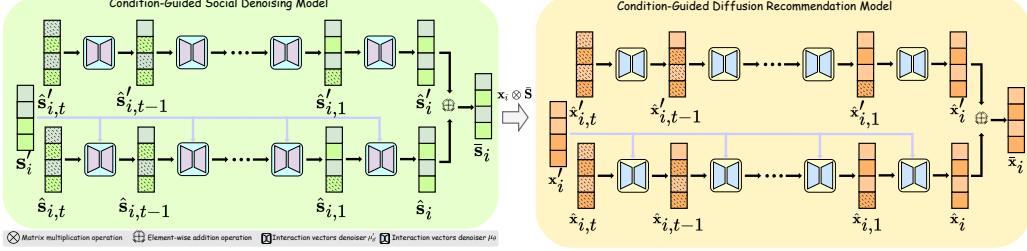


Figure 4: The Joint Inferencing process of the proposed method.

user-user co-purchasing long-tail items relationships as condition. By incorporating the social relationships of users who co-purchase long-tail items as conditions, CSD gains the capability to prioritize the social relationships between users who share preferences for the same long-tail items when removing redundant or noisy social relationships in social networks.

#### 4.4. Condition-Guided Diffusion Recommendation Model

##### 4.4.1. Adjusting the weights of items in social preferences

Instead of directly incorporating users' social preferences as conditions in the diffusion recommendation model, we aim to mitigate popularity bias by strategically adjusting the weights of long-tail and hot items within the social preferences. Based on social homophily theory [6], we first explore the user's social preference with matrix multiplication as:

$$\mathbf{R}^s = \mathbf{S}\mathbf{R}, \quad (12)$$

the element  $r_{ij}^s$  in  $\mathbf{R}^s$  denotes the sum of interactions from all social neighbors of user  $u_i$  to item  $v_j$ . The larger value of  $r_{ij}^s$  indicates that user  $u_i$  is more likely to be interested in item  $v_j$ . Then, we invert the social preference matrix  $\mathbf{R}^s$  to yield a matrix  $\mathbf{R}^{sdb}$  designed to mitigate popularity bias, which can be computed as follows:

$$r_{ij}^{sdb} = f(r_{ij}^s), \quad (13)$$

$$f(r_{ij}^s) = \begin{cases} 1/r_{ij}^s & \text{if } r_{ij}^s \neq 0 \\ 0 & \text{otherwise} \end{cases}, \quad (14)$$

where  $f(\cdot)$  is a function designed to adjust user social preferences, aimed at mitigating the popularity bias of the model.  $r_{ij}^{sdb}$  and  $r_{ij}^s$  are the elements located at row  $i$  and column  $j$  in  $\mathbf{R}^{sdb}$  and  $\mathbf{R}^s$ , respectively.

#### 4.4.2. Introducing social information as a condition into diffusion recommendation model

In order to take advantage of the rich information in social networks, we propose a Condition-Guided Diffusion Recommendation Model (CGD) to introduce the user's social preference as a condition in the diffusion recommendation model to improve the recommendation performance and mitigate the popularity bias. Specifically, we incorporate the user's social preference matrix  $\mathbf{R}^s$  into the user-item interaction matrix  $\mathbf{R}$  as condition, which can be formulated as:

$$\mathbf{R}' = \lambda \mathbf{J} \odot f(\mathbf{R}^s) + \mathbf{R}, \quad (15)$$

where  $\mathbf{J} \in \mathbb{R}^{m \times n}$  is the matrix whose elements are all 1.  $\odot$  denotes the element-wise product.  $\lambda$  is a coefficient to control the effect of the user's social preferences on the model's recommendation results, thus avoiding that the user's social preferences dominate the conditional recommendation results. Finally, we incorporate the user's social preference in each inferencing step of the inference process of the diffusion recommendation model as follows:

$$p_\theta(\mathbf{x}_{i,t-1} | \mathbf{x}_{i,t}, \mathbf{x}'_i) = \mathcal{N}(\mathbf{x}_{i,t-1}; \mu'_\theta(\mathbf{x}_{i,t}, \mathbf{x}'_i, t), \Sigma_\theta), \quad (16)$$

$$\mu'_\theta(\mathbf{x}_{i,t}, \mathbf{x}'_i, t) = (1 - \gamma)\mu_\theta(\mathbf{x}_{i,t}, t) + \gamma\mu_\theta(\mathbf{x}'_i, t), \quad (17)$$

where  $\mu'_\theta(\mathbf{x}_{i,t}, \mathbf{x}'_i, t)$  and  $\Sigma_\theta = \Sigma_\theta(\mathbf{x}_{i,t}, t)$  are the Gaussian parameters.  $\mathbf{x}'_i = [x'_{i1}, x'_{i2}, \dots, x'_{in}]$  is the row vector in the interaction matrix  $\mathbf{R}'$  with the condition added. Significantly, the hyper-parameter  $\gamma$  is employed in our CGD to regulate the degree in the inferencing process integrating the user's social preferences. Through the implementation of the aforementioned approach, our model effectively refers users' social preferences during the item recommendation process.

#### 4.5. Model Training

In generally, the loss function  $\mathcal{L}_{CGD}$  of the Condition-Guided Diffusion Recommendation Model (CGD) can be divided into two components [19]: the denoising matching loss function  $\mathcal{L}_t$  and the reconstruction loss function  $\mathcal{L}_1$ , which can be calculated by:

$$\mathcal{L}_1 = \mathbb{E}_{q(\mathbf{x}_{i,1} | \mathbf{x}_{i,0})} [\|\hat{x}_\theta(\mathbf{x}_{i,1}, 1) - \mathbf{x}_{i,0}\|_2^2], \quad (18)$$

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{x}_{i,t} | \mathbf{x}_{i,0})} \left[ \frac{1}{2} \left( \frac{\bar{\alpha}_{t-1}}{1 - \bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \right) \|\hat{x}_\theta(\mathbf{x}_{i,t}, t) - \mathbf{x}_{i,0}\|_2^2 \right], \quad (19)$$

---

**Algorithm 1** The Joint Inference of CGSoRec

---

**Input:** The user-user interaction vectors  $\mathbf{s}_i$  of user  $u_i$ , the user-item interaction vectors  $\mathbf{x}_i$  of user  $u_i$ , the optimised parameter  $\theta'$  and  $\theta$ .

**Output:** The interaction vector  $\bar{\mathbf{x}}_i$  for user  $u_i$ .

- 1: Sample  $\epsilon \sim \mathcal{N}(0, I)$ .
  - 2: Obtain  $\mathbf{s}'_i$  via Eq. 8 and Eq. 9.
  - 3: Obtain  $\mathbf{s}_{i,T}, \mathbf{s}'_{i,T}$  via Eq. 2.
  - 4: **while**  $t = T, \dots, 1$  **do**
  - 5:   Calculate  $\hat{\mathbf{s}}_{i,t-1}$  and  $\hat{\mathbf{s}}'_{i,t-1}$  with  $\hat{\mathbf{s}}_{i,t}, \mathbf{s}'_i, \hat{\mathbf{s}}'_{i,t}$  and  $\mu_{\theta'}(\cdot)$  via Eq. 11.
  - 6: **end while**
  - 7: Obtain the final denoising result  $\bar{\mathbf{s}}_i$  via Eq. 20.
  - 8: Obtain  $\mathbf{x}'_i$  with via Eq. 15.
  - 9: Obtain  $\mathbf{x}_{i,T}, \mathbf{x}'_{i,T}$  via Eq. 2.
  - 10: **while**  $t = T, \dots, 1$  **do**
  - 11:   Calculate  $\hat{\mathbf{x}}_{i,t-1}$  and  $\hat{\mathbf{x}}'_{i,t-1}$  with  $\hat{\mathbf{x}}_{i,t}, \mathbf{x}'_i, \hat{\mathbf{x}}'_{i,t}$  and  $\mu_\theta(\cdot)$  via Eq. 17.
  - 12: **end while**
  - 13: Obtain the final recommendation result  $\bar{\mathbf{x}}_i$  via Eq. 21.
  - 14: **return** recommendation result  $\bar{\mathbf{x}}_i$
- 

where  $\hat{x}_\theta(\mathbf{x}_{i,t}, t)$  is the neural network with learnable parameters  $\theta$  predicting  $\mathbf{x}_{i,0}$  based on  $\mathbf{x}_{i,t}$ . We can optimize the parameter  $\theta$  in  $\hat{x}_\theta(\mathbf{x}_{i,t}, t)$  by minimizing  $\mathcal{L}_{CGD} = \sum_{t=1}^T \mathcal{L}_t$ . The loss function  $\mathcal{L}_{CSD}$  of the Condition-Guided Social Denoising Model (CSD) is similar to that of CGD.

#### 4.6. Joint Inferencing

To effectively mitigate the popularity bias inherent in recommendation model, we design a joint inference between the Condition-Guided Diffusion Recommendation Model (CGD) and the Condition-Guided Social Denoising Model (CSD) to obtain the final recommendation results. The overall process of joint inferencing is illustrated in Figure 4. Specifically, with CSD, we first acquire the corrupted user social interaction vector  $\mathbf{s}_{i,t}$  and corrupted user social interaction vector  $\mathbf{s}'_{i,t}$ . Then, we execute a  $t$ -step denoising operation on the corrupted interaction vector  $\mathbf{s}_{i,t}, \mathbf{s}'_{i,t}$  and introduce the interaction vector  $\mathbf{s}'_i$  as condition in each denoising step of  $\mathbf{s}_{i,t}$  to guide the denoising result via the neural networks  $\mu_{\theta'}$  in Eq. 11. At last, we balance the relation

between  $\hat{\mathbf{s}}'_i$  and  $\hat{\mathbf{s}}_i$  by hyper-parameter  $w_s$ :

$$\bar{\mathbf{s}}_i = (1 - w_s)\hat{\mathbf{s}}_i + w_s\hat{\mathbf{s}}'_i, \quad (20)$$

where  $\hat{\mathbf{s}}_i = (1 - \eta)\mu_{\theta'}(\mathbf{s}_{i,1}, 1) + \eta\mu_{\theta'}(\mathbf{s}'_i, 1)$ . We use  $\bar{\mathbf{s}}_i$  for user ranking and derive the ultimate denoising result of social relation for user  $u_i$ . Then, we apply a similar approach as CSD to obtain the corrupted user-item interaction vector  $\mathbf{x}_{i,t}$  and  $\mathbf{x}'_{i,t}$  and execute a  $t$ -step denoising operation on them via the neural networks  $\mu_\theta$  in Eq. 17 to obtain the recommendation results  $\hat{\mathbf{x}}'_i$  and  $\hat{\mathbf{x}}_i$ . And we also balance the relation between  $\hat{\mathbf{x}}'_i$  and  $\hat{\mathbf{x}}_i$  by hyper-parameter  $w_r$ , which can be formulated as:

$$\bar{\mathbf{x}}_i = (1 - w_r)\hat{\mathbf{x}}_i + w_r\hat{\mathbf{x}}'_i, \quad (21)$$

where  $\hat{\mathbf{x}}_i = (1 - \gamma)\mu_\theta(\mathbf{x}_{i,1}, 1) + \gamma\mu_\theta(\mathbf{x}'_i, 1)$ . The joint inference procedure of our proposed model is summarized in Algorithm 1.

#### 4.7. Model Analysis

Let  $m$  denote the number of users and  $n$  the number of items. In the joint inference phase, CGSoRec utilizes two conditional diffusion models (CSD and CGD) to provide unbiased recommendations to users. At each time step  $t$ , CSD requires  $O(m)$  to denoise the social relationships of users, while CGD requires  $O(n)$  to generate unbiased recommendations for each user based on the denoised social relationships. The time complexity of CGSoRec depends solely on the number of users, items, and time steps  $T$ . Therefore, the overall time complexity of CGSoRec during the inference phase is  $O(T(m + n))$ . In contrast, traditional recommendation models typically have a time complexity of  $O(mn)$  with  $m, n \gg T$  in most cases, because they require the computation of similarity between each user and all items. This implies that CGSoRec is more scalable to larger datasets compared to traditional methods.

## 5. Experiments

In this section, we perform comprehensive experiments on three real-world datasets over three classic social recommendation models and our proposed CGSoRec to answer the following questions (RQs):

- **(RQ1)** How does the integration of social networks impact the performance and popularity bias of the **CGSoRec** compared to existing recommendation methods?

- **(RQ2)** How significant is the performance improvement for both hot and long-tail items when **CGSoRec** compared with **DiffRec** and **MACR**?
- **(RQ3)** What are the contributions of different components of **CG-SoRec** to performance?
- **(RQ4)** How do the hyper-parameters affect the performance of the proposed method **CGSoRec**?

### 5.1. Experimental Settings

#### 5.1.1. Datasets

To evaluate the effectiveness of the proposed method, we performed comprehensive experiments on three representative recommendation datasets with social information, including LastFM [48], DBook [49], and Ciao [13]. The statistical summary of these three datasets is shown in Table 1.

- **LastFM** [48]. This dataset is sourced from last.fm<sup>1</sup>, a UK-based Internet radio and music community. It is particularly valuable for testing social recommendation models due to its rich social network data. Each user in the dataset is associated with a list of their favorite artists, along with the corresponding play counts. Additionally, the dataset provides user application tags, which can be leveraged to create content vectors. In total, this dataset comprises 1,853 users and 2,698 items.
- **DBook** [49]. This dataset is sourced from Douban.Book<sup>2</sup>, a recommendation system dataset compiled from the Douban website. It includes user ratings of books on a scale from 1 to 5, along with data on social connections among users. This dataset is the largest dataset of the three datasets considered in this paper, featuring 10,864 users and a collection of 22,346 distinct items.
- **Ciao** [13]. This dataset is provided by a product review website called Ciao<sup>3</sup>. It includes user ratings for items they have purchased and social connections between users. Since we focus on implicit feedback, we convert detailed ratings into binary values of 0 or 1, indicating whether

---

<sup>1</sup><https://www.last.fm>

<sup>2</sup><https://book.douban.com>

<sup>3</sup><http://www.ciao.com>

the user is interested in the item or not. The version of this dataset used in our proposed method contains 5,868 users and 10,724 items.

- Gowalla [34]: This dataset is collected from LightGCN<sup>4</sup>, where users share their locations through check-ins. To maintain data quality, we retains only users and items with at least ten interactions. To incorporate social networks, we simulate a social network based on co-purchase relationships in this dataset, where each user retains a subset of users with frequent co-purchases as their friends.

Table 1: Statistics of the experimental data.

Dataset	LastFM	DBook	Ciao	Gowalla
#of users	1,853	10,864	5,868	29,858
#of items	2,698	22,346	10,724	40,981
#of Interactions ( <b>R</b> )	46,542	787,286	143,217	1,027,370
#of Density ( <b>R</b> )	0.00931	0.00324	0.00228	0.00084
#of Connections ( <b>S</b> )	25,435	169,151	111,782	/
#of Density ( <b>S</b> )	0.00741	0.00143	0.00335	/

### 5.1.2. Evaluation metrics

We employ two widely recognized metrics to assess the performance of top-K recommendations. Here,  $K$  represents the number of recommendations provided to users by the model. While a larger  $K$  can yield more stable evaluation results, it may not accurately reflect the targeted quality of the recommendations from a user’s perspective. In our experiments, we set  $K$  to 5 and 10. Higher values of these metrics indicate superior performance of the model in making recommendations. The mathematical definitions of these evaluation metrics are as follows:

- **Recall@K.** This metric assesses the model’s capability to recommend a list of items to users:

$$Recall@K = \frac{\sum_{u_i \in U} |R(u_i) \cap T(u_i)|}{\sum_{u_i \in U} |T(u_i)|} \quad (22)$$

---

<sup>4</sup><https://github.com/gusye1234/LightGCN-PyTorch>

where  $R(u_i)$  represents the list of recommendations made to the user, derived from their interactions in the training set,  $T(u_i)$  corresponds to the user's interactions in the testing set.

- **NDCG@K.** This metric evaluates the quality of the recommendations by considering positional importance of each item in the recommended list:

$$DCG_{u_i} @ K = \sum_{(u_i, v_j) \in D^{\text{test}}} \frac{\mathbf{I}(\hat{Z}_{u_i, v_j} \leq K)}{\log(\hat{Z}_{u_i, v_j} + 1)} \quad (23)$$

$$NDCG@K = \frac{1}{|U|} \sum_{u_i \in U} \frac{DCG_{u_i} @ K}{IDCG_{u_i} @ K} \quad (24)$$

where  $IDCG_{u_i} @ K$  is the ideal  $DCG_{u_i} @ K$ .  $\hat{Z}_{u_i, v_j}$  denotes the rank position of a positive feedback  $(u_i, v_j)$ .  $D^{\text{test}}$  is the testing set.

### 5.1.3. Parameter Setting

In the training process of our proposed method, we set the parameters as: the batch size is set to 400 as with DiffRec. The hidden layer size of the model  $d$  is searched in the range  $\{[300], [200, 600], [1000], [1500]\}$ . We choose the learning rates  $lr$  in  $\{0.0001, 0.0005, 0.001, 0.005\}$  of our proposed method, and perform a grid search to find the best parameters. We divide the interactions into train, test, and valid set with a ratio of  $8 : 1 : 1$ . The specific experiment is described below. We use Adam to optimize the model's parameters. For a fair comparison, we closely follow the default setting of the baselines.

### 5.1.4. Dataset splitting

It is important to note that when dealing with randomly sliced recommendation datasets, both training and test sets exhibit long-tail distributions. This can lead to improved results even if the model's popularity bias is intensified, consequently masking the adverse effects of popularity bias on the recommendation model. To address this, we draw on previous works [11, 2] to construct a debiased testing set where the interactions are sampled to be a uniform distribution over items, providing a more accurate measure of model performance. Specially, the better the model performs on the debiased test

set, the lower the popularity bias, and vice versa. This allows us to compare popularity bias across models using Recall and NDCG on the debiased test set.

#### 5.1.5. Baseline models

To evaluate the effectiveness of **CGSoRec**, we compare the proposed method with representative social recommendation models, including the graph-based social recommendation models (**DiffNet** [12], **LightGCN-S** [34]), the self-supervised learning-based social recommendation model (**MHCN** [1]), diffusion recommendation model(**DiffRec** [19]) and the debiased recommendation model( **MACR** [11]). These baseline methods are summarized as follows:

- DiffNet [12]: This is a GCN-based social recommendation method that models the recursive dynamic social diffusion in both user and item spaces. We omit the multiplication operation between the user feature matrix and the user-user interaction matrix at each layer of the model to eliminate the influence of the social network.
- LightGCN-S: This is an enhanced model, building upon LightGCN [34], that integrates social network information to boost the performance of recommendation systems. We exclude the user-user interaction matrix from the adjacency matrix to remove the influence of the social network.
- MHCN [1]: This is a self-supervised learning-based social recommendation method that integrates hypergraph convolutional networks. We omit the social self-supervised learning loss within the target loss function to eliminate the impact of social networks.
- GDMSR [2]: This is a social recommendation method which improves graph-based social recommendation by leveraging user-item feedback to filter noisy social relations and enhance preference learning, addressing the lack of reliably labeled data. We incorporate this method into MHCN to validate its effectiveness.
- DiffRec-S: This is an enhanced model based on DiffRec [19] that learns the generative process in a denoising manner to reconstruct the user-item interaction matrix. We incorporate unadjusted user social preferences as conditions into the training process of the diffusion model.

Table 2: Performance changes comparison of different social recommendation methods under a debiased testing set.

Basemodel	Datasets	LastFM				DBook				Ciao				Gowalla			
		Method	R@5	R@10	N@10	R@5	R@10	N@10	R@5	R@10	N@10	R@5	R@10	N@10	R@5	R@10	N@10
DiffNet	w/o S	Ori.	0.0178	0.0323	0.0343	0.0049	0.0077	0.0091	0.0026	0.0053	0.0048	0.0050	0.0102	0.0256			
		Ori.	0.0165	0.0313	0.0321	0.0045	0.0072	0.0084	0.0024	0.0049	0.0044	0.0046	0.0094	0.0250			
		Improv.	-7.30%	-3.10%	-6.41%	-8.16%	-6.49%	-7.69%	-7.69%	-7.54%	-8.33%	-8.00%	-0.78%	-2.34%			
LightGCN-S	w/o S	Ori.	0.0156	0.0334	0.0349	0.0044	0.0071	0.0079	0.0027	0.0054	0.0047	0.0047	0.0099	0.0258			
		Ori.	0.0151	0.0323	0.0344	0.0041	0.0069	0.0075	0.0025	0.0049	0.0044	0.0042	0.0091	0.0243			
		Improv.	-3.21%	-3.29%	-1.43%	-6.82%	-2.82%	-5.06%	-7.41%	-9.26%	-6.38%	-10.64%	-8.08%	-5.81%			
MHCN	w/o S	Ori.	0.0149	0.0328	0.0328	0.0034	0.0060	0.0066	0.0024	0.0042	0.0042	0.0053	0.0113	0.0253			
		Ori.	0.0148	0.0304	0.0301	0.0032	0.0055	0.0061	0.0022	0.0037	0.0038	0.0051	0.0107	0.0248			
		Improv.	-0.67%	-7.32%	-8.23%	-5.88%	-8.33%	-7.58%	-8.33%	-11.90%	-9.52%	-3.77%	-5.31%	-1.98%			
GDMSR	w/o S	Ori.	0.0149	0.0328	0.0328	0.0034	0.0060	0.0066	0.0024	0.0042	0.0042	0.0053	0.0113	0.0253			
		Ori.	0.0146	0.0306	0.0296	0.0029	0.0052	0.0060	0.0023	0.0039	0.0040	0.0050	0.0109	0.0251			
		Improv.	-2.01%	-6.71%	-9.76%	-14.71%	-13.33%	-9.09%	-4.17%	-7.14%	-4.76%	-5.66%	-3.54%	-0.79%			
MACR-S	w/o S	Ori.	0.0241	0.0572	0.0454	0.0031	0.0052	0.0097	0.0034	0.0060	0.0052	0.0093	0.0192	0.0320			
		Ori.	0.0184	0.0488	0.0379	0.0030	0.0050	0.0094	0.0031	0.0057	0.0051	0.0081	0.0180	0.0295			
		Improv.	-23.65%	-14.69%	-16.52%	-3.23%	-3.85%	-3.09%	-8.82%	-5.00%	-1.92%	-5.38%	-6.25%	-7.81%			
DiffRec-S	w/o S	Ori.	0.0188	0.0405	0.0399	0.0061	0.0114	0.0139	0.0026	0.0052	0.0052	0.0051	0.0115	0.0263			
		Ori.	0.0171	0.0380	0.0365	0.0049	0.0090	0.0113	0.0025	0.0045	0.0046	0.0045	0.0097	0.0252			
		Improv.	-8.51%	-6.17%	-8.52%	-19.67%	-21.05%	-18.71%	-3.85%	-13.46%	-11.54%	-11.76%	-15.65%	-4.18%			
CGSoRec	w/o S	Ori.	0.0188	0.0405	0.0399	0.0061	0.0114	0.0139	0.0026	0.0052	0.0052	0.0051	0.0115	0.0263			
		Ori.	0.0227	0.0491	0.0442	0.0065	0.0121	0.0150	0.0031	0.0057	0.0057	0.0059	0.0131	0.0296			
		Improv.	20.74%	21.23%	10.78%	6.56%	6.14%	7.91%	19.23%	9.62%	9.62%	15.69%	13.91%	12.55%			

- MACR-S: This is a social network enhancement method based on MACR [11] that mitigates the popularity bias of recommendation models by causal learning. We incorporate user social networks into the adjacency matrix  $\mathbf{A}$ , thereby integrating the social networks into the MACR model.

### 5.2. Performance and popularity bias changes comparison (RQ1)

Table 2 and Fig. 5 reports the overall performance and popularity bias changes of compared models before and after the incorporation of social networks, separately. According to the results, we can have the following observations:

- Existing recommendation models including social network like DiffNet, MHCN, LightGCN-S and DiffRec show a decline in performance when social information is incorporated. This is primarily because the social network amplifies the popularity bias of the recommendation model. Consequently, the model tends to recommend hot items to users while reducing the frequency of long-tail items.
- Our proposed **CGSoRec** exhibits significant performance improvements across three datasets following the integration of social networks.

Table 3: Performance on hot and long tail item group under a debiased testing set.

Basemodel	Datasets	Group	Method	LastFM			DBook			Ciao		
				R@5	R@10	N@10	R@5	R@10	N@10	R@5	R@10	N@10
MACR-S	Hot	w/o S	0.1046	0.1517	0.1197	0.0259	0.0496	0.0283	0.0074	0.0182	0.0105	
		Ori.	0.1053	0.1619	0.1254	0.0301	0.0570	0.0328	0.0087	0.0191	0.0111	
		Improv.	0.67%	6.72%	4.76%	16.22%	14.92%	15.90%	17.57%	4.95%	5.71%	
	Tail	w/o S	0.0234	0.0517	0.0390	0.0026	0.0048	0.0091	0.0027	0.0058	0.0049	
		Ori.	0.0180	0.0462	0.0322	0.0023	0.0044	0.0086	0.0025	0.0054	0.0046	
		Improv.	-23.08%	-10.64%	-17.44%	-11.54%	-8.33%	-5.49%	-7.41%	-6.90%	-6.12%	
DiffRec-S	Hot	w/o S	0.1178	0.1936	0.1372	0.0703	0.1094	0.0761	0.0175	0.0318	0.0209	
		Ori.	0.0910	0.1827	0.1258	0.0611	0.0910	0.0646	0.0150	0.0299	0.0194	
		Improv.	-22.75%	-6.93%	-8.31%	-13.09%	-16.82%	-15.11%	-14.29%	-5.97%	-7.18%	
	Tail	w/o S	0.0167	0.0376	0.0323	0.0035	0.0069	0.0084	0.0017	0.0032	0.0028	
		Ori.	0.0146	0.0363	0.0303	0.0028	0.0054	0.0069	0.0014	0.0028	0.0024	
		Improv.	-12.57%	-3.46%	-6.19%	-20.00%	-21.74%	-17.86%	-17.65%	-12.5%	-14.29%	
CGSoRec	Hot	w/o S	0.1178	0.1936	0.1372	0.0703	0.1094	0.0761	0.0175	0.0318	0.0209	
		Ori.	0.0940	0.1865	0.1227	0.0666	0.1035	0.0748	0.0166	0.0308	0.0202	
		Improv.	-20.20%	-3.67%	-10.57%	-5.26%	-5.39%	-1.71%	-5.14%	-3.14%	-3.35%	
	Tail	w/o S	0.0167	0.0376	0.0323	0.0035	0.0069	0.0084	0.0017	0.0032	0.0028	
		Ori.	0.0210	0.0461	0.0366	0.0038	0.0076	0.0094	0.0023	0.0037	0.0032	
		Improv.	25.75%	22.61%	13.31%	8.57%	10.14%	11.90%	35.29%	15.63%	14.29%	

For instance, on the LastFM dataset, Recall@5 increased from 0.0188 to 0.0227, an enhancement exceeding 20%. These improvements demonstrate the efficacy of our social denoising and the adjustment of user social preference weights.

- The performances of MHCN and GDMSR are lower than that of LightGCN across three datasets. This could be attributed to the hypergraph convolution module in them, which intensifies the model’s popularity bias. As a result, even after removing the self-supervised loss, the popularity bias of MHCN and GDMSR remains higher than that of LightGCN.
- The performance of the debiased social enhancement method MACR-S surpasses that of our proposed method, CGSoRec, on certain datasets. This is attributed to MACR’s use of causal learning techniques to reduce popularity bias in recommendation models. However, when social networks are incorporated into the MACR model, its performance experiences a notable decline. In contrast, the performance of our CGSoRec model improves upon the integration of social networks, which effectively demonstrates the efficacy of our approach.
- All baseline methods (DiffNet, LightGCN-S, MHCN, MACR-S and DiffRec-S) exhibit an increased recommendation frequency for hot items

following the introduction of the social network. This suggests that the social network indeed amplifies the popularity bias in the recommendation model. In contrast, our proposed model **CGSoRec** demonstrates a decrease in the recommendation frequency of hot items following the introduction of social networks. This is attributed to our innovative approach of social denoising and the adjustment of users' social preference weights.

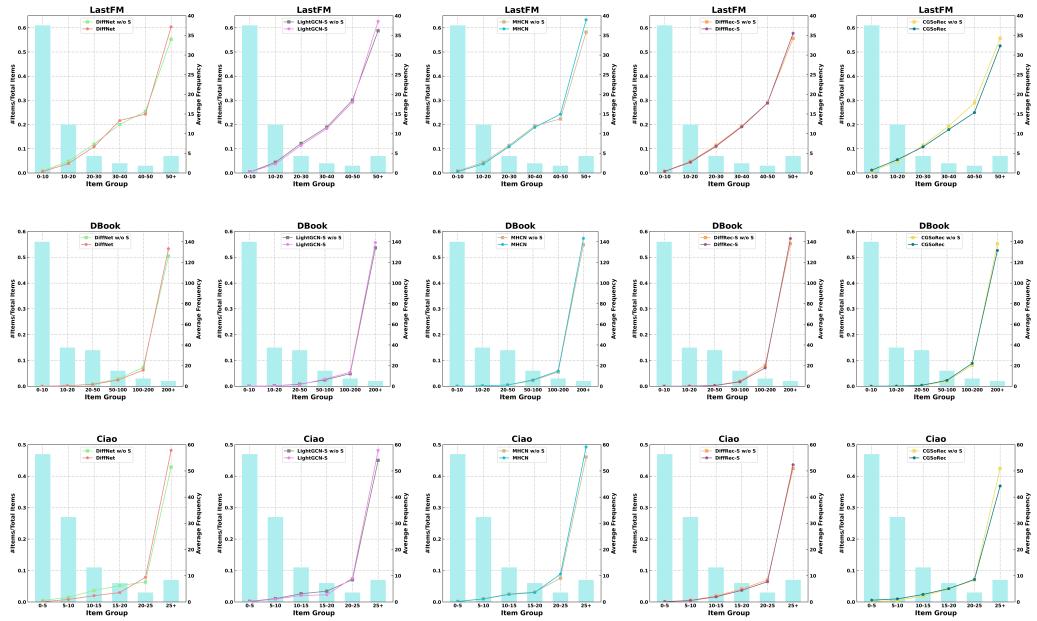


Figure 5: Frequency of different item groups recommended by DiffNet, LightGCN-S, MHCN and CGSoRec before and after the incorporation of social networks in LastFM and Ciao dataset.

### 5.3. Performance comparison in hot and long-tail items (RQ2)

In this subsection, we examine the performance improvements of the proposed model achieved for both hot and long-tail items. Based on the number of interact (Top 2% - 7% frequently interacted and rest) in the training set, the item groups are divided into the hot item group and the long-tail item group.

We present the recommendation performance of two distinct item groups on the LastFM, DBook and Ciao datasets under MACR-S, DiffRec-S and

our proposed method CGSoRec in Table 3. From the Table 3, we can see that with the introduction of social networks, the debiased recommendation method MACR enhances recommendation performance for hot items while decreasing performance for long-tail items. The primary reason for this phenomenon is that the social network makes MACR tend to recommend more hot items and fewer long-tail items to users. In addition, we observe that with the integration of social networks, the diffusion recommendation method DiffRec experiences a decline in recommendation performance for both hot and long-tail items. This is mainly due to the DiffRec method directly incorporates users' social preferences into the diffusion process as a condition, which is highly susceptible to noise within the social network. Consequently, even though the social network allows DiffRec to recommend more popular items, the noise in the network may impair the accuracy of these recommendations, thereby reducing the recommendation performance for hot items. Compared to DiffRec, our proposed method CGSoRec, upon incorporating social networks, trades a minor decline in recommendation performance for hot items for a significant improvement in the recommendation performance of long-tail items. As evident from the experimental results, CGSoRec traded a marginal reduction (3.67% under the Recall@10 metric on dataset Lastfm and 1.17% under the NDCG@10 metric on dataset DBook) in the recommendation performance of hot items for a substantial improvement (22.61% under the Recall@10 metric on dataset Lastfm and 11.90% under the NDCG@10 metric on dataset DBook) in the recommendation performance of long-tail items. This suggests that CGSoRec can effectively balance recommendation performance between hot and long-tail items and successfully mitigate the model's popular bias through social denoising and the adjustment of users' social preference weights.

#### 5.4. Ablation Studies

##### 5.4.1. Effect of Social Denoising and Weight Adjustment (RQ3)

To further study the effect of the Social Denoising and Weight Adjustment, we design two variants of our proposed **CGSoRec**: (1) **CGSoRec-SD**, where Condition-Guided Social Denoising Model (CSD) has been removed, and (2) **CGSoRec-SD&WA**, where Social Denoising and Weight Adjustment both have been removed. The experimental results are given in Table 4. We can see that Removing Social Denoising leads to a slight decrease in the model's performance. For example, in the LastFM dataset, after removing Social Denoising, the model's NDCG@10 decreases about

Table 4: Effect of Social Denoising and Weight Adjustment.

Datasets		LastFM			Dbook			Ciao		
Basemodel	Method	R@5	R@10	N@10	R@5	R@10	N@10	R@5	R@10	N@10
CGSoRec	Ori.	0.0227	0.0491	0.0442	0.0065	0.0121	0.0150	0.0031	0.0057	0.0057
	w/o SD	0.0218	0.0476	0.0435	0.0062	0.0118	0.0144	0.0029	0.0054	0.0055
	w/o SD&WA	0.0171	0.0380	0.0365	0.0049	0.0090	0.0113	0.0025	0.0045	0.0046

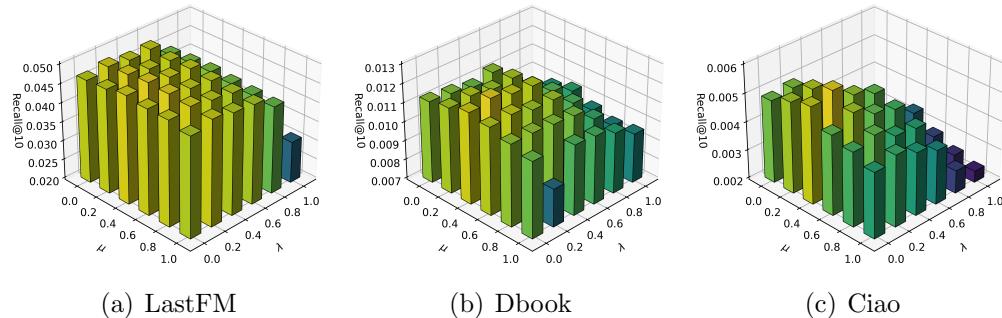


Figure 6: Effect of  $\mu$  and  $\lambda$  on CGSoRec w.r.t Recall@10.

1.6%. This maybe due to Social Denoising removing redundant social relationships in social networks, allowing the model to more accurately reflect users' social preferences. In contrast, the removal of Weight Adjustment has a more significant impact on the model's performance. For example, in the Dbook dataset, after removing Weight Adjustment, the model's Recall@10 decreases about 23.73%. This could be because Weight Adjustment mitigates the model's popularity bias by adjusting the weights of long-tail and popular items in users' social preferences. Removing Weight Adjustment deprives the model of its ability to debias the recommendation results, thus leading to a substantial decrease in the model's performance.

#### 5.4.2. Effect of hyper-parameter $\mu$ and $\lambda$ (RQ4)

In this subsection, we analyze the impact of the hyper-parameters  $\mu$  and  $\lambda$  on the model performance. Figure 6 shows the variation in model performance with respect to  $\mu$  and  $\lambda$  across three datasets, from which we observe that the choice of the two hyper-parameters is crucial for the performance of CGSoRec. Smaller values of  $\mu$  and  $\lambda$  often lead to better model performance. Overall, CGSoRec exhibits robustness to these two hyper-parameters on the LastFM dataset. Although the optimal choice of hyper-parameters

varies across different datasets, larger values of  $\mu$  and  $\lambda$  generally lead to a sharp decline in performance. This phenomenon occurs across all three datasets. In our experiments, we selected the optimal combination of these two hyper-parameters to achieve the best experimental results.

#### 5.4.3. Effect of hyper-parameter $\omega_r$ (RQ4)

The hyper-parameter  $\omega_r$  as formulated in Eq. 21 controls the degree to which the condition-guided recommendation result is involved in the final prediction. We execute a series of experiments on the LastFM, DBook and Ciao dataset using our proposed **CGSoRec** model, assessing its performance specifically in terms of NDCG@10. The experimental results are given in Fig. 7. We can see that, in lastFM and Ciao dataset, as the value of  $\omega_r$  varies from 0 to 0.5, the model’s performance progressively improves. However, further increasing  $\omega_r$  beyond this point is counterproductive. Similarly, in DBook dataset, as the value of  $\omega_r$  varies from 0 to 0.3, the model’s performance progressively improves. And further increasing  $\omega_r$  beyond this point is counterproductive. This observation indicates that the proper integration of conditional model recommendation results is advantageous in mitigating model popularity bias and enhancing overall model performance.

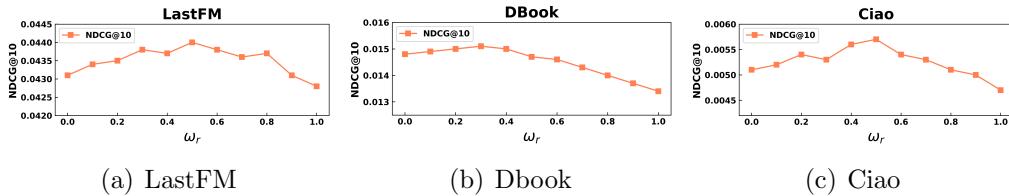


Figure 7: Effect of  $\omega_r$  on CGSoRec w.r.t NDCG@10.

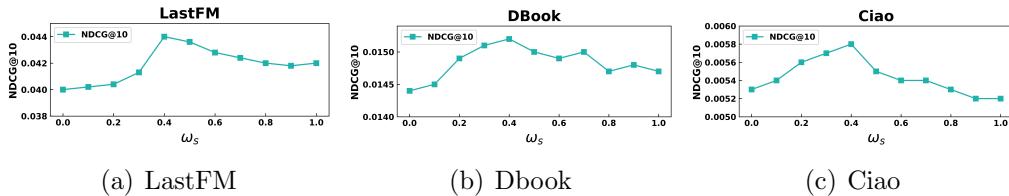


Figure 8: Effect of  $\omega_s$  on CGSoRec w.r.t NDCG@10.

#### 5.4.4. Effect of hyper-parameter $\omega_s$ (RQ4)

The hyper-parameter  $\omega_s$  as formulated in Eq. 20 controls the degree to which the condition-guided recommendation result is involved in the final denoised result of social network. A larger value of parameter  $\omega_s$  indicates that the social relationships between two users who have similar preferences for long-tail items are more likely to be preserved during the denoising process. Additionally, as  $\omega_s$  increases, the conditions will also dominate the final denoising result, leading to a decrease in the denoising capability of the model. We also execute a series of experiments on the LastFM, DBook and Ciao dataset using our proposed **CGSoRec** model, assessing its performance specifically in terms of NDCG@10. The experimental results are given in Fig. 8. In LastFM, DBook and Ciao dataset, as the value of  $\omega_s$  varies from 0 to 0.4, the model’s performance progressively improves. And when we further increase  $\omega_s$  beyond this point is counterproductive. This phenomenon suggests that appropriate conditions are also necessary to guide the denoising process of social networks, in order to obtain denoising results that are conducive to mitigate popularity bias in recommendation model.

## 6. Conclusion

In this work, we proposed a Condition-Guided Social Recommendation Model (**CGSoRec**), which can mitigate the popularity bias within the recommendation model by the adjusted social performance. Our model offers advanced components that yield two key advantages: (1) The Condition-Guided Social Denoising Model (CSD) included in **CGSoRec** can remove redundant or noisy social relations in social networks for more accurate user social preferences. (2) The Condition-Guided Diffusion Recommendation Model (CGD) included in **CGSoRec** can introduce the adjusted social performance (increasing the weight of long-tail items while decreasing the weight of hot items) as a condition to control the recommendation results. Comprehensive experiments conducted on three real-world datasets effectively demonstrate the efficacy of our proposed method. The ablation studies also demonstrate that the two components of our model can mitigate the model’s popularity bias and improve the recommendation performance.

## References

- [1] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, X. Zhang, Self-supervised multi-channel hypergraph convolutional network for social recommenda-

- tion, in: Proceedings of the web conference 2021, 2021, pp. 413–424.
- [2] Y. Quan, J. Ding, C. Gao, L. Yi, D. Jin, Y. Li, Robust preference-guided denoising for graph based social recommendation, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 1097–1108.
  - [3] B. Lacker, S. F. Way, Socially-motivated music recommendation, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 18, 2024, pp. 879–890.
  - [4] W. Ma, Y. Wang, Y. Zhu, Z. Wang, M. Jing, X. Zhao, J. Yu, F. Tang, Madm: A model-agnostic denoising module for graph-based social recommendation, in: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024, pp. 501–509.
  - [5] J. Wu, W. Fan, J. Chen, S. Liu, Q. Li, K. Tang, Disentangled contrastive learning for social recommendation, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 4570–4574.
  - [6] M. McPherson, L. Smith-Lovin, J. M. Cook, Birds of a feather: Homophily in social networks, *Annual review of sociology* 27 (1) (2001) 415–444.
  - [7] P. Zheng, Z. Huang, Y. Dou, Y. Yan, Rumor detection on social media through mining the social circles with high homogeneity, *Information Sciences* 642 (2023) 119083.
  - [8] X. Sha, Z. Sun, J. Zhang, Y.-S. Ong, Who wants to shop with you: Joint product–participant recommendation for group-buying service, *IEEE Transactions on Neural Networks and Learning Systems* (2022).
  - [9] H. Dong, W. Wang, K. Huang, F. Coenen, Automated social text annotation with joint multilabel attention networks, *IEEE Transactions on Neural Networks and Learning Systems* 32 (5) (2020) 2224–2238.
  - [10] W.-D. Xi, L. Huang, C.-D. Wang, Y.-Y. Zheng, J.-H. Lai, Deep rating and review neural network for item recommendation, *IEEE Transactions on Neural Networks and Learning Systems* 33 (11) (2021) 6726–6736.

- [11] T. Wei, F. Feng, J. Chen, Z. Wu, J. Yi, X. He, Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1791–1800.
- [12] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, in: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, 2019.
- [13] J. Tang, H. Gao, H. Liu, mtrust: Discerning multi-faceted trust in a connected world, in: Proceedings of the fifth ACM international conference on Web search and data mining, 2012, pp. 93–102.
- [14] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, D. Jin, Disentangling user interest and conformity for recommendation with causal embedding, in: Proceedings of the Web Conference 2021, 2021, pp. 2980–2991.
- [15] Y. Li, J. Liu, J. Ren, Social recommendation model based on user interaction in complex social networks, *PloS one* 14 (7) (2019) e0218957.
- [16] H. Orgad, B. Kawar, Y. Belinkov, Editing implicit assumptions in text-to-image diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 7053–7061.
- [17] K. Meng, D. Bau, A. Andonian, Y. Belinkov, Locating and editing factual associations in gpt, *Advances in Neural Information Processing Systems* 35 (2022) 17359–17372.
- [18] N. De Cao, W. Aziz, I. Titov, Editing factual knowledge in language models, *arXiv preprint arXiv:2104.08164* (2021).
- [19] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, T.-S. Chua, Diffusion recommender model, *arXiv preprint arXiv:2304.04971* (2023).
- [20] Z. Yang, J. Wu, Z. Wang, X. Wang, Y. Yuan, X. He, Generate what you prefer: Reshaping sequential recommendation via guided diffusion, *Advances in Neural Information Processing Systems* 36 (2024).
- [21] J. Walker, T. Zhong, F. Zhang, Q. Gao, F. Zhou, Recommendation via collaborative diffusion generative model, in: International Conference

on Knowledge Science, Engineering and Management, Springer, 2022, pp. 593–605.

- [22] J. Choi, S. Hong, N. Park, S.-B. Cho, Blurring-sharpening process models for collaborative filtering, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1096–1106.
- [23] Y. Tashiro, J. Song, Y. Song, S. Ermon, Csd: Conditional score-based diffusion models for probabilistic time series imputation, *Advances in Neural Information Processing Systems* 34 (2021) 24804–24816.
- [24] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, *Advances in neural information processing systems* 34 (2021) 8780–8794.
- [25] L. Zhang, A. Rao, M. Agrawala, Adding conditional control to text-to-image diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3836–3847.
- [26] J. Ho, T. Salimans, Classifier-free diffusion guidance, arXiv preprint arXiv:2207.12598 (2022).
- [27] J. Yu, Y. Wang, C. Zhao, B. Ghanem, J. Zhang, Freedom: Training-free energy-guided conditional diffusion model, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 23174–23184.
- [28] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, L. Cui, Enhancing social recommendation with adversarial graph convolutional networks, *IEEE Transactions on knowledge and data engineering* 34 (8) (2020) 3727–3739.
- [29] W. Wang, F. Feng, X. He, L. Nie, T.-S. Chua, Denoising implicit feedback for recommendation, in: Proceedings of the 14th ACM international conference on web search and data mining, 2021, pp. 373–381.
- [30] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: Proceedings of the fourth ACM conference on Recommender systems, 2010.
- [31] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: Proc. of WWW, 2019.

- [32] G. Guo, H. Zhou, B. Chen, Z. Liu, X. Xu, X. Chen, Z. Dong, X. He, Ipgan: Generating informative item pairs by adversarial sampling, *IEEE transactions on neural networks and learning systems* 33 (2) (2020) 694–706.
- [33] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, Q. Li, Deep adversarial social recommendation, in: 28th International Joint Conference on Artificial Intelligence, IJCAI 2019, 2019, pp. 1351–1357.
- [34] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.
- [35] L. Wu, H. Lin, C. Tan, Z. Gao, S. Z. Li, Self-supervised learning on graphs: Contrastive, generative, or predictive, *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [36] H.-S. Sheu, Z. Chu, D. Qi, S. Li, Knowledge-guided article embedding refinement for session-based news recommendation, *IEEE Transactions on Neural Networks and Learning Systems* 33 (12) (2021) 7921–7927.
- [37] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, L. Cui, Enhance social recommendation with adversarial graph convolutional networks, *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [38] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).
- [39] N. Li, B. Guo, Y. Liu, Z. Yu, Disentangled-feature and composite-prior vae on social recommendation for new users, *Expert Systems with Applications* 247 (2024) 123309.
- [40] L. Li, J. Xiahou, F. Lin, S. Su, Distvae: distributed variational autoencoder for sequential recommendation, *Knowledge-Based Systems* 264 (2023) 110313.
- [41] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, *Advances in neural information processing systems* 33 (2020) 6840–6851.

- [42] Y. Xuan, Diffusion cross-domain recommendation, arXiv preprint arXiv:2402.02182 (2024).
- [43] Y. Wang, Z. Liu, L. Yang, P. S. Yu, Conditional denoising diffusion for sequential recommendation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2024, pp. 156–169.
- [44] D. Liang, R. G. Krishnan, M. D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: Proceedings of the 2018 world wide web conference, 2018, pp. 689–698.
- [45] Z. Zhou, S. Tulsiani, Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 12588–12597.
- [46] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, M. Pavone, Guided conditional diffusion for controllable traffic simulation, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 3560–3566.
- [47] C. Luo, Understanding diffusion models: A unified perspective, arXiv preprint arXiv:2208.11970 (2022).
- [48] I. Cantador, A. Bellogín, D. Vallet, Content-based recommendation in social tagging systems, in: Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 237–240.
- [49] Y. Lu, Y. Fang, C. Shi, Meta-learning on heterogeneous information networks for cold-start recommendation, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1563–1573.