Mitigating Recommendation Biases via Group-Alignment and Global-Uniformity in Representation Learning

MIAOMIAO CAI, Hefei University of Technology, China
MIN HOU, Hefei University of Technology, China
LEI CHEN, Tsinghua University, China
LE WU*, Hefei University of Technology, China
HAOYUE BAI, Hefei University of Technology, China
YONG LI, Tsinghua University, China
MENG WANG*, Hefei University of Technology, China

Collaborative Filtering (CF) plays a crucial role in modern recommender systems, leveraging historical user-item interactions to provide personalized suggestions. However, CF-based methods often encounter biases due to imbalances in training data. This phenomenon makes CF-based methods tend to prioritize recommending popular items and performing unsatisfactorily on inactive users. Existing works address this issue by rebalancing training samples, reranking recommendation results, or making the modeling process robust to the bias. Despite their effectiveness, these approaches can compromise accuracy or be sensitive to weighting strategies, making them challenging to train. Therefore, exploring how to mitigate these biases remains in urgent demand.

In this paper, we deeply analyze the causes and effects of the biases and propose a framework to alleviate biases in recommendation from the perspective of representation distribution, namely ***Group-Alignment and Global-Uniformity Enhanced Representation Learning for Debiasing Recommendation (AURL)***. Specifically, we identify two significant problems in the representation distribution of users and items, namely group-discrepancy and global-collapse. These two problems directly lead to biases in the recommendation results. To this end, we propose two simple but effective regularizers in the representation space, respectively named group-alignment and global-uniformity. The goal of group-alignment is to bring the representation distribution of long-tail entities closer to that of popular entities, while global-uniformity aims to preserve the information of entities as much as possible by evenly distributing representations. Our method directly optimizes both the group-alignment and global-uniformity regularization terms to mitigate recommendation biases. Please note that *AURL* applies to arbitrary CF-based recommendation backbones. Extensive experiments on three real datasets and various recommendation backbones verify the superiority of our proposed framework. The results show that *AURL* not only outperforms existing debiasing models in mitigating biases but also improves recommendation performance to some extent.

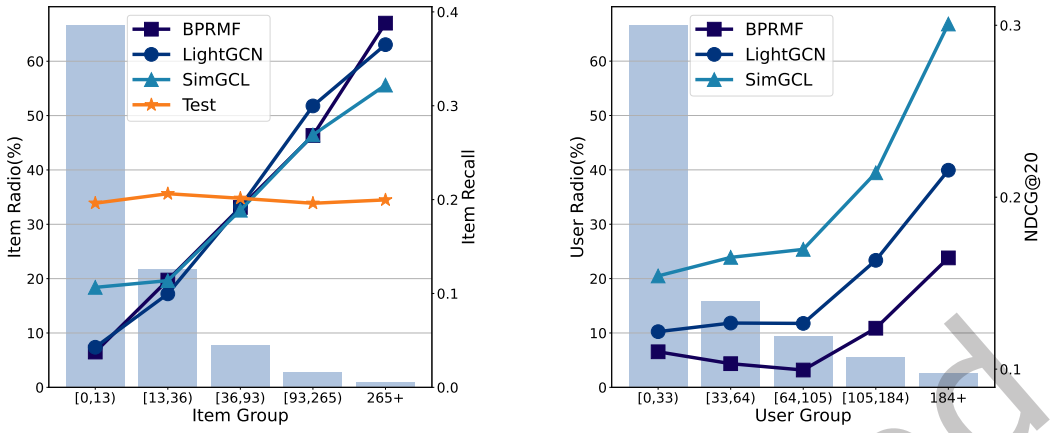
CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Collaborative Filtering, Representation Learning, Alignment, Uniformity

*Corresponding Authors.

Authors' addresses: Miaomiao Cai, Hefei University of Technology, Hefei, China, cmm.hfut@gmail.com; Min Hou, Hefei University of Technology, Hefei, China, hmhoumin@gmail.com; Lei Chen, Tsinghua University, Beijing, China, chenlei.hfut@gmail.com; Le Wu, Hefei University of Technology, Hefei, China, lewu.ustc@gmail.com; Haoyue Bai, Hefei University of Technology, Hefei, China, baihaoyue621@gmail.com; Yong Li, Tsinghua University, Beijing, China, liyong07@tsinghua.edu.cn; Meng Wang, Hefei University of Technology, Hefei, China, eric.mengwang@gmail.com.

ACM 2157-6912/2024/5-ART
https://doi.org/10.1145/3664931



(a) Item popularity bias (b) User consistency bias

Fig. 1. We analyze biases in the results of three typical CF models—BPRMF [33], LightGCN [18], and SimGCL [55] on the Douban-Book dataset [54]. To facilitate our illustration, we categorize items and users into groups based on their popularity in the training set. We then evaluate, based on the TopK recommendation lists, the recommendation frequency (*Item Recall*) for each item group and the accuracy performance (*NDCG*@20) for each user group.

1 INTRODUCTION

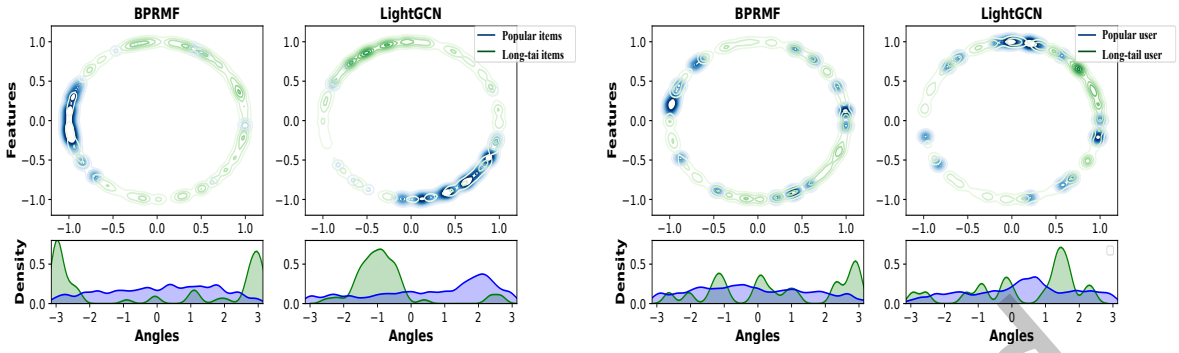
Personalized recommendations have become indispensable in various online applications, serving as valuable tools for users to cope with information overload [4, 20, 53]. As the most popular schema for personalized recommender systems, Collaborative Filtering (CF) [18, 33, 55] utilizes similarities between users and items hidden in historical user-item interactions to provide recommendations. Typically, CF-based methods encode users and items into a shared space and then recover the user-item interactions (preferences) through the corresponding representations [47].

Although CF-based methods have achieved considerable success, their training approach, which involves reconstructing historical interactions, makes them susceptible to imbalances within the training data, leading to biases in recommendation results [6, 47, 65]. In real-world scenarios, the frequency distribution of users and items in training interactions is uneven. They often follow a power-law distribution, where a small number of popular items and active users dominate the majority of interactions [47]. This phenomenon causes CF-based methods to prioritize recommending popular items and perform unsatisfactorily on inactive users, thereby failing to uncover the real characteristics of items and the true preferences of users [6]. Fig. 1 shows the item popularity bias and user consistency bias on the real-world Douban-Book dataset [54]. We group items and users according to their frequency (popularity) of appearance in the training data, and the background histograms indicate the ratio of items/users in each group [47]. We can observe a clear phenomenon of data imbalance. Then we train the mainstream CF-based models BPRMF [33], LightGCN [18] and SimGCL [55]. We count the frequency of items in the recommendation results for each item group (Fig. 1(a)) and calculate the recommendation performance for each user group (Fig. 1(b)). In Fig. 1(a), the orange line shows the real item frequency in the test dataset. As evident, items that are more popular in the training data are recommended far more frequently than anticipated, highlighting a significant item popularity bias. In Fig. 1(b), there is inconsistency in the effectiveness of the recommendations between different user groups, and inactive users experience unsatisfactory recommendations. These biases significantly impact the performance of recommender systems, undermining both the diversity of recommendations and the user experience [5, 6, 47]. Even more concerning, item popularity bias can cause the "Matthew effect", where popular items receive more recommendations and consequently become even more popular [47].

Given the significant impact of biases, debiasing in recommender systems has recently become a key area of research [5, 6, 10, 14, 59, 61, 65]. Previous efforts have addressed bias mitigation from several angles: (1) Sampling Strategies: Some studies down-sample [10] popular entities (items and users) or up-sample [65] unpopular ones to balance interaction distributions in the training set. (2) Robust Methods: Techniques such as causal inference [47, 63] and adversarial training [5, 50] have been employed to enhance model robustness against biased data [61, 65]. (3) Post-Processing: Certain approaches re-rank recommendation lists to prevent the over-recommendation of popular items[6, 65]. While these methods are commendable, they exhibit imperfections, such as potentially compromising recommendation accuracy by oversimplifying data balancing [6], which can ignore users' true preferences. Causal inference methods[47], requiring stringent data generation assumptions, and adversarial training [65], which introduces instability by blurring distinctions between popular and unpopular entities, illustrate the challenges of existing approaches [50, 63]. Therefore, further exploration into effective bias mitigation remains critically necessary.

In fact, it is known that the quality of learned representations plays a crucial role in the recommendation. Most CF-based recommendation models can be divided into two parts [51]. Firstly, encoders project users and items into a representation space [23]. Then, an interaction function computes preferences between users and items in the space [19]. The interaction function is usually set as the simple inner product, while researchers meticulously design various kinds of encoders to make the representations as informative as possible [41]. Consequently, we highlight that **addressing the impact of biases from the perspective of representation distribution and obtaining better representations** contains a huge potential to effectively resolve the issue of biases. However, achieving this goal is not trivial. Initially, scrutinizing the impact of biases on representations is imperative. In Fig.2, we respectively map the learned representations of two typical CF-based models as 2-dimensional normalized vectors on the unit hypersphere $\mathcal{S}^1$ (i.e., the circle with radius 1). Then, we plot the feature distribution using nonparametric Gaussian Kernel Density Estimation (KDE) in $\mathbb{R}^2$ and visualize the density estimations at angles for each entity on $\mathcal{S}^1$. According to Fig.2, we can observe notably different feature/density distributions between popular entities and long-tail entities. We believe that this phenomenon arises from the imbalanced data, which results in a scarcity of interactions for long-tail entities, thereby hindering the acquisition of accurate representations. Consequently, the distributions of representations for long-tail entities exhibit inconsistency when compared to those of popular entities with adequate interactions. Ideally, there should be no distributional shift between the popular entities and long-tail entities. We simplify this phenomenon as **group-discrepancy**. Furthermore, we find that the distribution of entity representations exhibits a folding pattern, clustering around several points. This clustering phenomenon indicates that the learned representations lack informativeness and fail to capture the distinctive characteristics of each entity [42]. We name the drawback as **global-collapse**. Addressing the two impacts of bias on representation distribution is the key to mitigating recommendation biases.

In this paper, we advocate for a paradigm shift in analyzing and addressing both item-side popularity bias and user-side consistency bias from the perspective of representation distribution. We begin by investigating the causes of group-discrepancy and global-collapse through both mathematical and empirical analyses. To address these two issues, we propose two simple but effective regularizers in representation space respectively, named group-alignment and global-uniformity. Specifically, the group-alignment regularizer aims to bring the representation distribution of long-tail entities closer to that of popular entities. This regularizer transfers knowledge from the well-trained representations of popular entities to those of long-tail entities, thereby enhancing the latter's representation quality. Inspired by the uniformity property in contrastive learning [42], we design a uniformity-based regularize. The regularizer leads the representations roughly uniformly distributed on the unit hypersphere, preserving as much information of the entities as possible. To this end, we propose a framework from the representation distribution, namely ***Group-Alignment and Global-Uniformity Enhanced Representation Learning for Debiasing Recommendation (AURL)***. Our methodology directly optimizes the group-alignment



(a) Item representation distribution (b) User representation distribution

Fig. 2. Representation distribution of the Douban-Book dataset on $\mathcal{S}^1$. We plot the representation distributions using Gaussian Kernel Density Estimation (KDE) in $\mathbb{R}^2$ and von Mises-Fisher (vMF) KDE on angles (i.e., $\arctan 2(y, x)$ for each point (x, y) on $\mathcal{S}^1$). Specifically, we categorize items and users into two groups based on their popularity: blue represents popular items/users, while green denotes unpopular items/users.

and global uniformity regularization terms to reduce group-discrepancy and global-collapse, thereby further mitigating bias issues in recommendation results. Extensive testing on three real-world datasets confirms the efficacy of our method in reducing biases and enhancing recommendation accuracy across various CF-based models. The main contributions of this paper are summarized as follows:

(1) We advocate a novel perspective, utilizing representation distribution, to address both item-side popularity bias and user-side consistency bias, supported by comprehensive mathematical analyses and empirical evidence.
(2) We design the group-alignment and global-uniformity regularizers to effectively counter the biases induced by group-discrepancy and global-collapse, respectively.
(3) Extensive testing on three real-world datasets verifies the efficacy of our method in reducing biases and improving recommendation accuracy across various CF-based models.

2 RELATED WORK

2.1 Collaborative filtering

Collaborative Filtering (CF) is a widely used technique in recommender systems, aiming to provide personalized recommendations by leveraging users' preferences and behaviors [33]. The basic idea behind CF is that users with similar interests and preferences are likely to have similar opinions about items [23]. MMost CF-based models can be divided into two components [51]. Firstly, encoders project users and items into a representation space [23]. Subsequently, an interaction function computes the preferences between users and items within this space. The interaction function is typically set as a simple inner product, while researchers meticulously design various encoders to make the representations as informative as possible [41].

The simplest encoders can directly map user and item IDs into the representation space [23, 33]. With the development of deep learning, neural-based encoders such as multi-layer perceptrons [19] and attention mechanisms [8] have emerged in recent years to capture the complex relationships between users and items. User-item interaction data can naturally be organized into a bipartite graph, prompting researchers to employ Graph Neural Networks (GNNs) [7, 12, 45, 48] to encode more accurate node representations and high-order structural information. For example, NGCF [45], LR-GCCF [9], and LightGCN [18] utilize high-order relationships on interaction graphs to enhance representation performance. Recently, Self-Supervised Learning (SSL) has been introduced to improve the generalizability of representations [49]. For example, SimGCL [55] employs noise

feature enhancement methods and constructs comparison targets to enhance the accuracy and robustness of representations. Despite their effectiveness, CF-based models often overlook biases in recommendation results due to the imbalanced distribution of interaction data [6].

2.2 Debiasing methods in recommendation results

Mitigating biases in recommendation results is a common task in recommender systems and has been extensively studied. Previous research work has tried to alleviate recommendation biases from multiple perspectives. We classify the methods into re-weighting-based, decorrelation-based, and adversarial-based approaches.

Re-weighting-based methods aim to shift attention away from popular items/users either during training or prediction, thereby increasing the importance of unpopular items/users in the recommendation process [5, 11, 25, 27, 61, 65]. For example, Inverse Propensity Scoring (IPS) [65] compensates for unpopular items/users by adjusting predictions within the user-item preference matrix, thereby elevating preference scores and rankings for unpopular items/users. Explanation: Expanded the acronym "IPS" for clarity and refined the sentence structure to improve readability. Similarly, γ-AdjNorm [61] enhances the focus on unpopular items/users by controlling the normalization strength during the neighborhood aggregation process in GCNs-based models. DORL [13] addresses the Matthew effect in offline reinforcement learning recommendations by introducing a penalty term, mitigating the conservatism inherent in existing methods. Zerosum [34] reduces model bias in recommendation systems by directly equalizing recommendation scores across items preferred by a user.

Decorrelation-based methods aim to mitigate the influence of popularity on item/user representations or prediction scores by removing correlations between them [2, 30, 44, 47, 50, 58]. For example, MACR [47] utilizes counterfactual reasoning to eliminate the direct impact of popularity on item/user outcomes. TIDE [62] leverages temporal information to differentiate between benign bias due to item quality and harmful bias resulting from conformity.

Adversarial-based methods aim to engage in a minimax game between the recommender G and an introduced adversary D, such that D provides signals to increase the recommendation opportunities for unpopular items or enhance user accuracy [24, 26, 36, 46, 50, 60]. FairGo [50] enhances the graph network recommendation model by incorporating discriminators, which predict fairness-related attributes of nodes by utilizing their embeddings and the embeddings of the surrounding network structure. FairMI [59] employs adversarial principles to minimize mutual information between embeddings and sensitive attributes while maximizing it between embeddings and non-sensitive information. In contrast, InvCF [56] exploits the notion that item/user representations remain unchanged despite variations in popularity semantics. By filtering out unstable or outdated popularity characteristics, InvCF learns unbiased representations.

Although previous methods have worked hard to reduce biases in recommendation results, they still have certain limitations. For example, sampling strategies and post-processing methods usually only focus on long-tail items/users, at the expense of accuracy for popular items/users [65]. Methods based on causal reasoning typically rely on strong assumptions about data generation [47], and adversarial learning methods are often unstable [50]. Therefore, there is still an urgent demand to explore more effective methods to address these biases in recommender systems [6].

2.3 Representation learning

Representation learning plays a crucial role in collaborative filtering by generating personalized representations for each user and item [52]. These personalized representations more accurately reflect the interests of users and the characteristics of items, leading to enhanced accuracy and effectiveness in CF-based models [18, 23, 31]. Without effective representation learning, models may struggle to capture the underlying relationships between users and items accurately, resulting in less accurate recommendations [49].

In the field of representation learning, researchers typically focus on two fundamental properties to measure the quality of learned representations: alignment and uniformity [42]. The goal of alignment is to ensure that the learned representations effectively capture the similarities between positive data points [42]. This enhances the model's ability to express relationships by bringing similar data points closer together in the representation space. In Natural Language Processing (NLP), alignment is used to align words or phrases between different languages, facilitating cross-lingual text translation and information retrieval [35]. In Computer Vision (CV), alignment is applied to synchronize features across images from different domains, such as aligning images of varied styles or sensors into a unified representation space for applications like image generation [66], style transfer [37], and more. In recommender systems (RS), alignment can be used to align user and item representations to better capture user interests and item characteristics [41]. Unlike previous studies, this research focuses on the relationship between popular entity groups and long-tail entity groups. By aligning the representation distributions of different groups, we aim to achieve improved group-alignment, ensuring that the distribution of entities in the representation space remains consistent, regardless of changes in popularity.

The goal of uniformity is to ensure that learned representations are evenly distributed in the feature space, typically by having these representations roughly evenly spread across the unit hypersphere [42, 57]. This enhances the model's generalization performance, leading to a more balanced and consistent distribution of representations [49, 55]. In NLP, uniformity ensures that word representations are evenly distributed within the semantic space, improving the quality of word vectors and thereby enhancing performance in text-related tasks [57]. In CV, uniformity ensures that feature representations are evenly distributed across the image space, helping the model more effectively capture diverse image features [57]. In RS, uniformity ensures that the distribution of user and item representations is more informative, thereby improving both the accuracy and generalization of recommendations [41, 49, 55].

Previous studies have demonstrated that InfoNCE optimizes both the alignment and uniformity of representations by aligning positive samples and distancing negative samples [42, 55]. However, these studies typically focus on aligning different views of the same user or item post data augmentation, which can introduce selection bias in terms of uniformity optimization. In contrast, methods such as DirectAU [41] directly optimize both alignment and uniformity, thus circumventing issues related to selection bias in positive and negative sample selection. In this paper, we explore the issue of inconsistent distributions across different item sets, emphasizing the need to align from the perspective of feature distributions. By integrating both alignment and uniformity, our approach effectively optimizes feature representations, enhances recommendation performance, and adapts better to variations in data distributions, thus delivering improved outcomes for recommendation systems.

3 IMPACT OF BIAS ON REPRESENTATION LEARNING IN CF

3.1 Debiasing problem formulation

In this subsection, we first establish a formal definition of the debiasing problem in CF. Let $\boldsymbol{U}$ (where $|\boldsymbol{U}| = M$) and $\boldsymbol{I}$ (where $|\boldsymbol{I}| = N$) denote the sets of users and items in the CF-based models, respectively. Assuming the implicit feedback setting, let $\mathcal{R} \in 0, 1^{M \times N}$ represent the observed implicit interaction matrix, where $\mathcal{R}_{u,i} = 1$ if user u has interacted with item i; otherwise, it is 0. The key to CF-based models is accurately learning the user representation matrix $\mathbf{Z} \in \mathbb{R}^{M \times D}$ and the item representation matrix $\mathbf{H} \in \mathbb{R}^{N \times D}$, where D denotes the representation size and $D \ll M, N$. With the representation matrix, the predicted score is defined as the similarity between the user and item representation. Specifically, the similarity between users and items is calculated using the inner product of their representations, i.e., $s(u, i) = \mathbf{z}_u^T \mathbf{h}_i$. Here $s(u, i)$ is the prediction score of user u to the item i, $\mathbf{z}_u$ and $\mathbf{h}_i$ denote the representation of user u and item i. To directly capture information from interactions, most studies employ the Bayesian Personalized Ranking (BPR) loss [33], a meticulously designed ranking objective function

for recommendations. Formally, the BPR loss is as follows:

$$\mathcal{L}_{BPR} = -\frac{1}{|\mathcal{R}|} \sum_{(u,i)\in\mathcal{R}, i^- \in \boldsymbol{I}/\boldsymbol{I}_u^+} ln\sigma(s(u,i) - s(u,i^-)), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function, $\boldsymbol{I}_u^+$ represents the set of items interacted by users in the training, i is the positive item that the user has interacted with, and i^- is a randomly sampled negative item that the user has not interacted with. Specifically, the BPR loss ensures that the predicted score of observed interactions is higher than that of sampled unobserved interactions.

In this paper, focusing on the CF-based recommendation task, we investigate biases in recommendation results, specifically addressing user-side consistency bias and item-side popularity bias. To better define and analyze biases in recommendation results, we divide items and users into two groups based on their popularity levels, namely:

$$\boldsymbol{U} = G_{pop}^{user} \cup G_{tail}^{user}, \boldsymbol{I} = G_{pop}^{item} \cup G_{tail}^{item}, \tag{2}$$

where G_{pop}^{user} and G_{pop}^{item} represent the popular user and item groups, respectively, while G_{tail}^{user} and G_{tail}^{item} denote the long-tail user and item groups. All groups are mutually exclusive. In our work, we aim to ensure that all groups have fair performance. Specifically, for user consistency bias, we expect the model to provide similar recommendation quality across user groups, regardless of their popularity levels. We define the debiasing objective on the user side by Demographic Parity (DP) as provided in [6] :

$$\mathbb{E}_{u\in G_{pop}^{user}}[ACC(u)] = \mathbb{E}_{u\in G_{tail}^{user}}[ACC(u)], \tag{3}$$

where $ACC(u)$ define the recommendation accuracy of user u. In contrast to user-side bias, for item popularity bias, we aim for each item group to have equal opportunities for exposure in the TopK recommendations, formally stated as:

$$p(G_{pop}^{item}|TopK) = p(G_{tail}^{item}|TopK), \tag{4}$$

where $p(G_{pop}^{item}|TopK)$ and $p(G_{tail}^{item}|TopK)$ respectively represent the probability of popular item and long-tail item being in the TopK recommendation list.

In summary, the primary objective of this paper is to mitigate biases in recommendation results, aiming to ensure fairness across different entity groups and reduce disparities in those results. Additionally, we aim to preserve the competitive advantage of the recommendation model while minimizing any adverse effects on its overall effectiveness.

3.2 Biases in representation distribution

In this subsection, we conduct an in-depth analysis of the causes and manifestations of the biases in representation distribution. We take the most popular BPR loss as an example to analyze the impact of biases. Formally, for a training sample $(u,i) \in \mathcal{R}$, the BPR loss is defined as:

$$\mathcal{L}_{(u,i,i^-)} = -ln\sigma(s(u,i) - s(u,i^-)) = -ln\sigma(\mathbf{z}_u^T\mathbf{h}_i - \mathbf{z}_u^T\mathbf{h}_{i^-}),\ where\ i^- \in \boldsymbol{I}/\boldsymbol{I}_u^+. \tag{5}$$

To optimize the BPR loss function using Stochastic Gradient Descent (SGD) [1], we calculate the gradients of the user representation $\mathbf{z}u$, positive sample representation $\mathbf{h}i$, and negative sample representation $\mathbf{h}i^-$ as follows:

$$\nabla_{z_u} = \frac{\partial\mathcal{L}_{(u,i,i^-)}}{\partial\mathbf{z}_u} = -(1-\sigma(s(u,i)-s(u,i^-)))(\mathbf{h}_i - \mathbf{h}_{i^-}), \tag{6}$$

$$\nabla_{\mathbf{h}_i} = \frac{\partial\mathcal{L}_{(u,i,i^-)}}{\partial\mathbf{h}_i} = -(1-\sigma(s(u,i)-s(u,i^-)))\mathbf{z}_u, \tag{7}$$

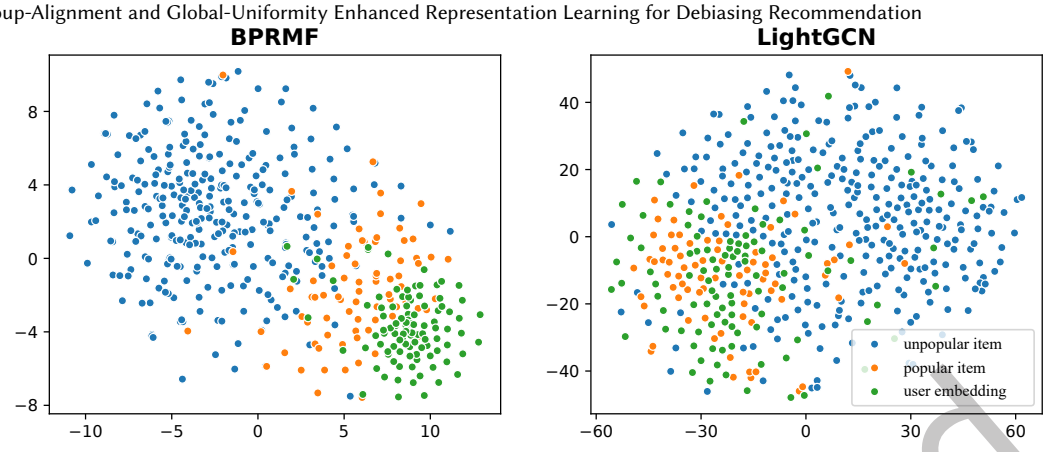


Fig. 3. Representation visualization of items and users in the Douban-Book dataset. We randomly selected 500 items and 200 users and utilized T-SNE to visualize the representation spaces of BPRMF and LightGCN, respectively. In the visualization, green dots represent users, while blue and orange dots represent popular and long-tail items, respectively. It is evident that the number of long-tail items significantly exceeds that of popular items.

$$\nabla_{\mathbf{h}_{i^-}} = \frac{\partial \mathcal{L}_{(u,i,i^-)}}{\partial \mathbf{h}_{i^-}} = (1 - \sigma(s(u,i) - s(u,i^-)))\mathbf{z}_u. \tag{8}$$

According to these equations, it is evident that the gradient update directions for the positive sample representation $\mathbf{h}i$ and the negative sample representation $\mathbf{h}i^-$ are diametrically opposite. As recommendation data typically follows the power-law distribution, positive items are often popular items sampled from $(u,i) \sim \mathcal{R}$. Conversely, negative samples are usually drawn randomly from the entire itemset $\boldsymbol{I}$, typically resulting in long-tail items. Consequently, when using SGD to optimize the BPR loss, popular items and long-tail items tend to be updated to distinct positions within the representation space:

$$\mathbf{h}_i \leftarrow \mathbf{h}_i - \eta \nabla_{\mathbf{h}_i}, \ \mathbf{h}_{i^-} \leftarrow \mathbf{h}_{i^-} - \eta \nabla_{\mathbf{h}_{i^-}}, \tag{9}$$

where η is the learning rate, and typically $i \in Gpop^{item}$ and $i^- \in G^{item}_{tail}$. As a result, the distribution of item groups in the representation space becomes inconsistent. Furthermore, it is observed that the update direction for users aligns with that of the positive sample, leading most users to cluster near popular items while distancing from long-tail items.

To intuitively understand the impact of biases on representation distribution, we visualize the representations of users and items using two common CF-based models, BPRMF [33] and LightGCN [18]. We map the learned representations into a two-dimensional (2D) space using t-SNE [39], ensuring all representations are captured at the point of optimal performance. From Fig. 3, we observe a distinctly different distribution of users and items. Consistent with our analysis, popular items and long-tail items are located in separate regions of the representation space, with user representations predominantly clustering around popular items. The phenomenon of inconsistent distribution may be the reason why the results of different groups are slightly different. This distribution pattern biases the model towards recommending popular items over considering users' actual preferences and the attributes of the items.

Furthermore, the discussion above highlights that the optimization directions for different entities are all aligned with the representations of positive entities. The frequent appearance of popular entities in the training set homogenizes the optimization direction of the representations, meaning they are optimized in similar directions. As illustrated in Fig. 2, representations from different groups tend to cluster around several focal points. This clustering indicates that the differences between representations are minimal, rendering the representations less

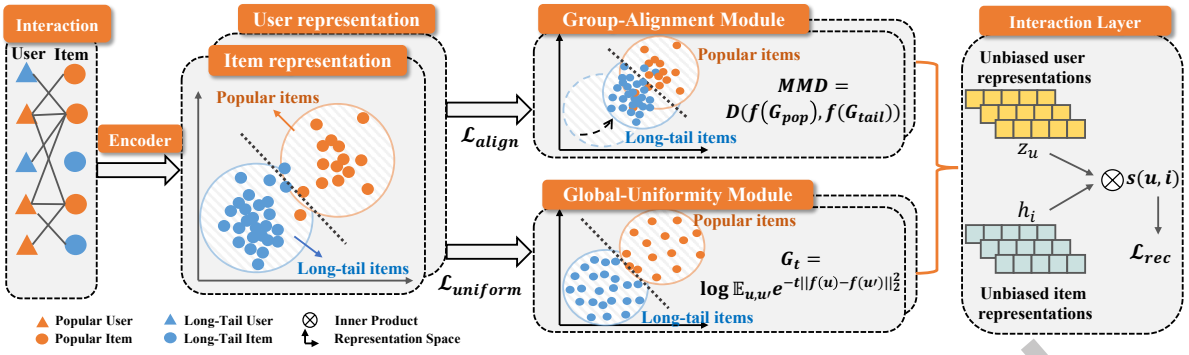


Fig. 4. An illustration of the *AURL* framework begins with input data being encoded through CF-based models to map users and items into the representation space. Subsequently, these representations are constrained by two modules: group-alignment $\mathcal{L}align$ and global-uniformity $\mathcal{L}uniform$, which work together to generate unbiased representations. Finally, the interaction function utilizes these unbiased representations to predict scores for user-item pairs, $s(u, i)$, as part of the recommendation task $\mathcal{L}_{rec}$. It is important to note that while the diagram specifically focuses on item-side debiasing, similar operations are conducted on the user side as well.

informative. Furthermore, this clustering causes the representations of different groups to diverge markedly, further exacerbating the bias towards popular entities. Based on the foregoing discussion, we summarize the impact of bias on representations as follows:

- **Group-Discrepancy**: The representations of various groups' entities are localized to distinct regions within the representation space, indicating a segregation based on group characteristics.
- **Global-Collapse**: The distribution of user and item representations shows a folding pattern, with the data densely clustering around a few focal points, leading to reduced representational diversity and potential loss of information.

4 THE PROPOSED MODEL

We introduce a framework, ***Group-Alignment and Global-Uniformity Enhanced Representation Learning for Debiasing Recommendation (AURL)***, designed to mitigate biases in recommender systems through representation distribution. The group-alignment module minimizes differences between popular and long-tail entities, thereby ensuring representation alignment. Inspired by contrastive learning [42], our global-uniformity module aims to enhance the quality of representations. The *AURL*framework is illustrated in Fig. 4. This section details the design and analysis of these modules, both theoretically and empirically, and presents the overall objective function.

4.1 Group-Alignment module

In the previous section, we demonstrated that CF-based methods result in biased distributions in the latent space, termed "group-discrepancy". Consequently, despite similarities, items and users are dispersed based on popularity, leading to a model bias towards popular entities. To address group-discrepancy, we propose group-alignment, aiming for popular and long-tail entities to share distribution characteristics. Due to data abundance and inherent user preference biases, popular entities' representations align more with user preferences, whereas long-tail entities' representations may be less precise. Our goal is to minimize the distributional distance between these entity groups in the representation space, thus standardizing representation distributions across groups. This involves transferring insights from well-established representations of popular entities to enhance those of long-tail entities. We formally define the optimality of group-alignment as follows:

Definition (Perfect Group-Alignment). The representation distribution is perfect group-alignment if the distribution of the popular entity group G_{pop} aligns perfectly with the distribution of the long-tail entity group G_{tail}, i.e., $p(f(G_{pop})) = p(f(G_{tail}))$. Here $p(f(\cdot))$ represents the distribution of representations for different groups. Drawing inspiration from domain adaptation techniques, we aim to minimize the distributional distance between two groups, thereby achieving group-aligned representation distributions. The formal representation of this concept is as follows:

$$\min_{\theta} \hat{D}(f(G_{pop}), f(G_{tail})), \tag{10}$$

where $\hat{D}(\cdot, \cdot)$ is an estimate of the distribution discrepancy between the representation of popular user/item groups $f(G_{pop})$ and that of unpopular user/item groups $f(G_{tail})$. Potential measures for this discrepancy include KL divergence [21], Maximum Mean Discrepancy (MMD) [38], among others.

Both KL divergence and Maximum Mean Discrepancy (MMD) are commonly used to measure the difference between two probability distributions. However, KL divergence requires knowledge of the probability density functions involved and is sensitive to the specific forms and assumptions underlying these distributions [21, 64]. In contrast, MMD offers several advantages when aligning distributions. MMD effectively measures differences between two distributions without assuming specific forms [38]. This is achieved by mapping distributions into a high-dimensional feature space and computing inner products between samples in this space to quantify differences. This approach captures overall distribution characteristics effectively, regardless of distribution shapes [29]. Furthermore, MMD eliminates the need for intermediate density estimation, resulting in more stable optimization and avoiding the challenges and instabilities associated with complex alternative optimization procedures [40]. Additionally, by directly maximizing the difference between distributions, MMD reduces sensitivity to initialization. This guidance helps models learn better feature representations and avoid suboptimal solutions. Overall, using MMD facilitates better alignment of different distributions and yields favorable outcomes in tasks like domain adaptation and unsupervised learning.

To achieve group alignment, we employ Maximum Mean Discrepancy (MMD)[38] as our regularizer to estimate the discrepancy between the two groups. MMD functions as a kernel-based two-sample test that assesses the null hypothesis $p(f(G_{pop})) = p(f(G_{tail}))$, based on the observed samples, where the encoder $f(\cdot)$ maps users/items into the representation space [16]. The fundamental concept of MMD is that if the generating distributions are identical, then all statistical measures derived from these distributions should also be identical [64]. Formally, MMD quantifies the following difference measures:

$$D_{\mathcal{H}}(G_{pop}, G_{tail}) =: \|\mathbb{E}[\phi(f(G_{pop}))] - \mathbb{E}[\phi(f(G_{tail}))]\|^2_{\mathcal{H}}, \tag{11}$$

where $\mathcal{H}$ represents the reproducing kernel Hilbert space (RKHS) equipped with a characteristic kernel k [64]. Here, $\phi(\cdot)$ denotes the feature map of the original representation to RKHS, and the kernel k is defined as $k(f(G_{pop}), f(G_{tail})) =< \phi(f(G_{pop})), \phi(f(G_{tail})) >$, where $< \cdot, \cdot >$ denotes the inner product of vectors. The central theoretical result is that $p(f(G_{pop})) = p(f(G_{tail}))$ holds if and only if $D_{\mathcal{H}}(G_{pop}, G_{tail}) = 0$ [16]. In practice, an estimate of the MMD involves comparing the squared distance between the empirical kernel mean representations, denoted as:

$$D^{\sim}_{\mathcal{H}}(G_{pop}, G_{tail}) =: \|\frac{1}{|G_{pop}|} \sum_{x_i \in f(G_{pop})} \phi(x_i) - \frac{1}{|G_{tail}|} \sum_{x_j \in f(G_{tail})} \phi(x_j)\|^2_{\mathcal{H}}, \tag{12}$$

where $D^{\sim}_{\mathcal{H}}(G_{pop}, G_{tail})$ is an unbiased estimator of $D_{\mathcal{H}}(G_{pop}, G_{tail})$ [64]. We use Eqn. (12) as the estimate of the discrepancy between G_{pop} and G_{tail}.

Based on our previous discussion, both the user and item sides exhibit the issue of group-discrepancy. Therefore, we simultaneously apply MMD to both the user and item sides as follows:

$$\begin{aligned}\mathcal{L}_{align} &= \frac{1}{2} \times (\mathcal{L}_{align}^{user} + \mathcal{L}_{align}^{item}) \\ &= \frac{1}{2} \times (D_{\mathcal{H}}^{2}(f(G_{pop}^{user}), f(G_{tail}^{user})) + D_{\mathcal{H}}^{2}(f(G_{pop}^{item}), f(G_{tail}^{item}))) \\ &= \frac{1}{2} \times (\| \frac{1}{|G_{pop}^{user}|} \sum_{x_i \in f(G_{pop}^{user})} \phi(x_i) - \frac{1}{|G_{tail}^{user}|} \sum_{x_j \in f(G_{tail}^{user})} \phi(x_j) \|_{\mathcal{H}}^{2} \\ &+ \| \frac{1}{|G_{pop}^{item}|} \sum_{x_i \in f(G_{pop}^{item})} \phi(x_i) - \frac{1}{|G_{tail}^{item}|} \sum_{x_j \in f(G_{tail}^{item})} \phi(x_j) \|_{\mathcal{H}}^{2}),\end{aligned} \tag{13}$$

where $|\cdot|$ represents the size of the set. It should be noted that the traditional training paradigm has effectively learned the representation of popular user/item groups. However, if the distance between the two group distributions is artificially reduced, there is a risk that the performance of the popular group may diminish. Therefore, we fix the representations of the popular entities and unilaterally push the representation distribution of long-tail entities closer to that of popular entities. As illustrated by the black arrow in Fig. 4, when we apply $\mathcal{L}_{align}$, the long-tail group (depicted with a blue dotted line) in the bias distribution moves closer to the popular group (depicted with an orange dotted line) in the representation space. This operation allows the model to enhance the long-tail entity group without compromising the performance of the popular entity group.

4.2 Global-Uniformity module

In addressing the global-collapse issue, we draw inspiration from the uniformity property observed in contrastive learning [42] to develop a global-uniformity regularizer. This regularizer aims to enhance the even distribution of representation across various users and items, which we refer to as global-uniformity. Optimal global-uniformity means that feature vectors are spread as uniformly as possible across the unit hypersphere $\mathcal{S}^{m-1}$, thereby preserving a higher degree of informational content. We now proceed to define the criteria for optimal global-uniformity in the representation distributions within collaborative filtering as follows [41]:

Definition (Perfect global-uniformity). The representation distribution is perfect global-uniformity if the distribution of $f(u)$ for $u \backsim p(\boldsymbol{U})$ and the distribution of $f(i)$ for $i \backsim p(\boldsymbol{I})$ are the uniform distribution σ_{d-1} on $\mathcal{S}^{d-1}$. Here, $\mathcal{S}^{d-1} = \{x \in \mathbb{R}_d : \|x\| = 1\}$ represents the surface of the $d - dimenaional$ unit sphere, and $p(\boldsymbol{U})$ and $p(\boldsymbol{I})$ denote the distributions of users and items, respectively. The global-uniformity property ensures that each representation preserves as much intrinsic information about the user or item as possible. Studies have demonstrated that improved global-uniformity enhances the quality of representations both theoretically and empirically [41, 42].

To implement a global-uniformity optimizer, we aim for the global-uniformity property to be both asymptotically correct (i.e., the distribution optimized by this metric should converge to a uniform distribution) and empirically reasonable with a finite number of points, as described in [42]. To achieve this, we utilize the Gaussian potential kernel (also known as the Radial Basis Function (RBF) kernel), $G_t : \mathcal{S}^d \times \mathcal{S}^d \to \mathbb{R}_+$ [3]:

$$G_t(f(v), f(v')) \triangleq e^{-t\|f(v)-f(v')\|_2^2} = e^{2t \cdot f(v)^T f(v') - 2t}, t > 0, \tag{14}$$

where v and v' represent any user or item, and t is a fixed parameter. We define the global-uniformity loss in the recommendation system as the logarithm of the average pairwise Gaussian potential on both the user side and

item side:

$$\begin{aligned}\mathcal{L}_{uniform} &= \frac{1}{2} \times (\mathcal{L}^{user}_{uniform} + \mathcal{L}^{item}_{uniform}) \\ &= \frac{1}{2} \times (log \mathop{\mathbb{E}}_{u,u^{'} \backsim p(U)} G_t(u,u^{'}) + log \mathop{\mathbb{E}}_{i,i^{'} \backsim p(I)} G_t(i,i^{'})) \\ &= \frac{1}{2} \times (log \mathop{\mathbb{E}}_{u,u^{'} \backsim p(U)} e^{-t\|f(u)-f(u^{'})\|_2^2} + log \mathop{\mathbb{E}}_{i,i^{'} \backsim p(I)} e^{-t\|f(i)-f(i^{'})\|_2^2}).\end{aligned} \tag{15}$$

It is worth noting that the perfectly uniform lower bound here is not a fixed value, but is dependent on the dimension of the representation space [42]. The above formula demonstrates that $\mathcal{L}_{uniform}$ encourages users and items to be distributed as evenly as possible across the entire unit sphere. This promotes more effective utilization of semantic information within the space. Maintaining a uniform distribution of feature vectors maximizes the retention of the original feature information in the data.

4.3 Analyses of Group-Alignment and Global-Uniformity

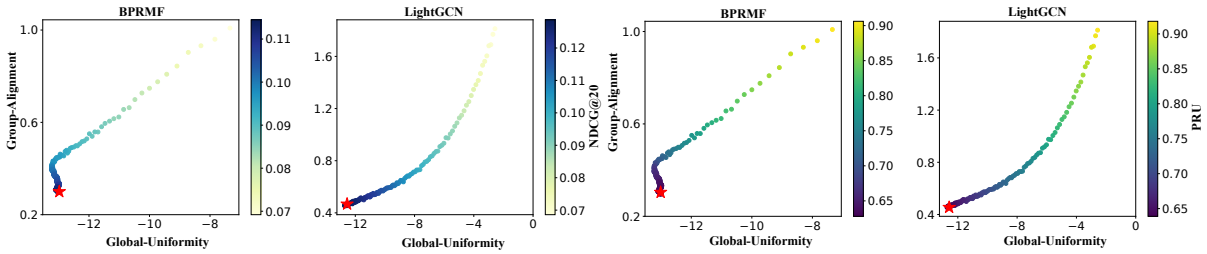
4.3.1 ***Theoretical Analyses.*** We first explore the necessity of group-alignment in mitigating the item-side popularity bias. We define $\mathcal{L}_{debias}$ as the target for debiasing under ideal conditions, which aims to equalize the expected preferences of users towards both the popular and long-tail item groups. Formally, $\mathcal{L}_{debias}$ for a user u is defined as:

$$\begin{aligned}\min_{\theta} \mathcal{L}_{debias} &= \min_{\theta} \mathbb{E}_{i \in G^{item}_{pop}}[s(u,i)] - \mathbb{E}_{i \in G^{item}_{tail}}[s(u,I)] \\ &= \min_{\theta} \mathbb{E}_{i \in G^{item}_{pop}}[\sigma(\mathbf{z}_u^T f(i))] - \mathbb{E}_{i \in G^{item}_{tail}}[\sigma(\mathbf{z}_u^T f(i))] \\ &= \min_{\theta} \sigma(\mathbf{z}_u^T(\mathbb{E}[f(G^{item}_{pop})] - \mathbb{E}[f(G^{item}_{tail})])) \\ &\propto \min_{\theta} \mathbb{E}[f(G^{item}_{pop}) - f(G^{item}_{tail})] \\ &= \min_{\theta} \sum_{i \in G^{item}_{pop}} f(i)p(f(G^{item}_{pop})) - \sum_{i \in G^{item}_{tail}} f(i)p(f(G^{item}_{tail})) \\ &= \min_{\theta} \sum_{i} f(i)[p(f(G^{item}_{pop}) - p(f(G^{item}_{tail}))]\end{aligned} \tag{16}$$

According to the definition of perfect group-alignment, Eqn. (16) tends towards zero only when the encoder achieves perfect group-alignment between the distributions of the two item groups. This suggests that reducing the distance between the distributions of the two item groups can help narrow the gap between the scores of the popular item group and the long-tail item group.

We then explore the necessity of group-alignment in mitigating the user-side consistency bias. Let $\mathbb{E}[\mathcal{L}_{G^{user}_{pop}}]$ and $\mathbb{E}[\mathcal{L}_{G^{user}_{tail}}]$ represent the expected value of the loss for the popular user group and the long-tail user group, respectively. The Bayesian Personalized Ranking (BPR) loss can then be reformulated as:

$$\begin{aligned}\mathcal{L}_{BPR} &= \frac{1}{|\mathcal{R}|} \sum_{(u,i) \in \mathcal{R}, i^- \in I/I^+_u} \mathcal{L}_{(u,i,i^-)} = \frac{1}{|\mathcal{R}|} \sum_{u \in |U|} \sum_{i \in I^+_u, i^- \in I/I^+_u} \mathcal{L}_{(u,i,i^-)} \\ &= \frac{1}{|\mathcal{R}|}(\sum_{u \in G^{user}_{pop}} \sum_{i \in I^+_u, i^- \in I/I^+_u} \mathcal{L}_{(u,i,i^-)} + \sum_{u \in G^{user}_{tail}} \sum_{i \in I^+_u, i^- \in I/I^+_u} \mathcal{L}_{(u,i,i^-)}) \\ &= \frac{1}{|\mathcal{R}|}(|\mathcal{R}^+_{G^{user}_{pop}}| \times \mathbb{E}[\mathcal{L}_{G^{user}_{pop}}] + |\mathcal{R}^+_{G^{user}_{tail}}| \times \mathbb{E}[\mathcal{L}_{G^{user}_{tail}}])\end{aligned} \tag{17}$$



(a) NDCG@20 (b) PRU

Fig. 5. Metrics and performance of BPRMF and LightGCN experiments are visualized. Each point on the plot represents a trained encoder, with its x and y coordinates indicating group-alignment and global-uniformity properties, respectively. The color of each point denotes validation set accuracy, measured by *NDCG*@20, and debiasing metrics, represented by *PRU*. Stars on the plot indicate the points where convergence was achieved.

where $\mathcal{R}^+_{G^{user}_{pop}}$ and $\mathcal{R}^+_{G^{user}_{tail}}$ respectively represent the interaction sets of popular users and long-tail users in the training set. Due to imbalanced data, the number of interactions for popular users ($|\mathcal{R}^+_{G^{user}_{pop}}|$) significantly exceeds that of long-tail users ($|\mathcal{R}^+_{G^{user}_{tail}}|$). This imbalance causes the training process to optimize predominantly for popular users, achieving a lower average loss, while the impact on long-tail users remains minimal. Consequently, the model tends to learn the characteristics and preferences of popular users more effectively, thereby optimizing the recommendation results for this group more successfully. To enhance the accuracy for long-tail users, our objective is to balance the loss between the two user groups, expressed as $\mathbb{E}[\mathcal{L}_{G^{user}_{pop}}] - \mathbb{E}[\mathcal{L}_{G^{user}_{tail}}]$. The proposed group-alignment property facilitates the transfer of knowledge from the well-trained representations of popular users to long-tail users, improving the representational distribution and learning of preferences for the latter.

Please note that achieving distribution consistency can be readily accomplished by encoding all items or users into the same representation; however, this approach risks inducing global-collapse [41]. To circumvent this issue, we have designed a global-uniformity regularizer. This regularizer ensures that representations are uniformly distributed across the unit hypersphere, thereby preserving maximal information about the entities.

4.3.2 ***Empirical Observations.*** To further investigate the relationship between the two properties and recommendation results, we depicted the variations of the properties, along with the accuracy of recommendations (*NDCG*@20) and the debias metrics (*PRU*), during the training process (Fig. 5). Lower values of group-alignment and global-uniformity are desirable. A lower *PRU* metric signifies less bias towards popularity in recommendations, while a higher *NDCG*@20 indicates better recommendation accuracy. The stars in Fig. 5 represent the convergence points of the model.

We observe a highly consistent behavior between the properties and the recommendation metrics. The upper two subfigures illustrate the trends of *NDCG*@20 with group-alignment and global-uniformity, where darker colors represent better recommendation quality. It is noticeable that smaller values of global-uniformity and group-alignment are more favorable for *NDCG*@20 improvement. The lower two subfigures in Fig. 5 showcase the relationship between the debias indicator *PRU* and representation distribution properties. Darker colors signify lower *PRU* values, indicating better debiasing performance. As evident, achieving better group-alignment or global-uniformity can both help alleviate the biases of the recommendation results and improve the accuracy of the model, and it may be beneficial to optimize them simultaneously.

In summary, our theoretical and empirical analyses demonstrate a strong correlation between the ideal debiasing objectives and the two properties we proposed. In other words, better group-alignment and global-uniformity are advantageous for both improving the accuracy of the model's recommendations and mitigating bias.

4.4 Objective function

The loss function of our framework consists of three components: recommendation loss ($\mathcal{L}_{rec}$), group-alignment loss ($\mathcal{L}_{align}$), and global-uniformity loss ($\mathcal{L}_{uniform}$). For details, $\mathcal{L}_{rec}$ represents the loss function for the recommendation task in any CF-based model. This could be, for example, the BPR loss [33], InfoNCE loss [55], Softmax loss [51], or any other suitable loss function used in collaborative filtering models. To prevent overfitting, we also apply L_2 regularization to constrain all the parameters of the model. The overall loss function is formulated as follows:

$$\mathcal{L}_{AURL} = \mathcal{L}_{rec} + \lambda_1 \mathcal{L}_{align} + \lambda_2 \mathcal{L}_{uniform} + \lambda \|\theta\|_F^2, \tag{18}$$

where λ_1 and λ_2 are hyperparameters that control the strengths of auxiliary tasks, $\theta = \{\mathbf{Z}, \mathbf{H}\}$ denotes all trainable model parameters, and λ governs the L_2 regularization strength to prevent overfitting. Additionally, it is important to note that the auxiliary tasks do not introduce additional parameters. In our framework, by jointly training $\mathcal{L}_{align}$ and $\mathcal{L}_{uniform}$, we achieve unbiased representations that help debias recommendation results. *AURL* is designed to alleviate biases in results from both the item side and the user side. Furthermore, *AURL* is compatible with arbitrary CF-based models by integrating two auxiliary loss functions.

4.5 Discussion

In this work, we primarily investigate the issue of popularity bias in recommendation system results and propose the ***Group-Alignment and Global-Uniformity Enhanced Representation Learning for Debiasing Recommendation (AURL)*** . We address both item-side popularity bias and user-side consistency bias by regularizing the distribution of representations, detailing the model of *AURL*. Specifically, we elaborate on the mechanism behind the group alignment regularizer, which aims to mitigate group discrepancies by aligning the distribution of long-tail entities with that of popular entities in the representation space. Additionally, we discuss the global uniformity regularizer, which combats global collapse phenomena by encouraging representations of entities on the hypersphere to be more uniformly distributed.

While our method has shown some effectiveness in addressing bias in recommendation systems, it has limitations. Our approach, which involves aligning and uniformizing the entire item/user space, might not effectively capture subtle differences, highlighting the need for more fine-grained grouping methods. Additionally, despite achieving group alignment, the method's underlying mechanism lacks intuitiveness and interpretability, which could hinder understanding and communication with users and stakeholders. These aspects suggest significant areas for further research and improvement in the interpretability of the method.

5 EXPERIMENTS

5.1 Experimental settings

5.1.1 ***Dataset.*** We conduct experiments on three public real-world datasets: Amazon-Book for e-commerce recommendation [55], Movielens-20M for movie recommendation [17], and Douban-Book for book recommendation [55]. Following previous experimental setups [18, 55], we retain only users and items with at least 5 interactions. For each user, 70% of interactions are randomly selected as the training set, another 10% as the validation set for hyperparameter tuning, and the remaining 20% as the test set.

To define user and item groups, we apply the "Pareto Principle", selecting the top 20% most frequently interacted users/items in the training set as the popular group, and the others as the long-tail group [28]. Detailed data statistics are summarized in Table 1.

5.1.2 ***Evaluation metrics.*** In this work, we place greater emphasis on the biases in model results rather than on accuracy. Therefore, we adopt two accuracy metrics to ensure that *AURL* does not significantly compromise

Table 1. Detailed datasets statistics.

Datasets	#Users	#Itmes	#Interactions	Density
Amazon-Book	52,643	91,599	2,984,108	0.0619%
Movielens-20M	99,626	14,387	28,011,110	0.1954%
Douban-Book	12,859	22,294	792,062	0.2087%

the model's accuracy, and we introduce two metrics to demonstrate how effectively *AURL* mitigates biases. To measure the accuracy of the recommendation results, we utilize Hit Ratio (*HR@K*) and Normalized Discounted Cumulative Gain (*NDCG@K*) [18]. To evaluate the debiasing capability of the model, we assess it from the perspectives of both users and items.

For the item side, we select Popularity-Rank Correlation (*PRU*) [61, 65], which measures the correlation between the user's true preference and the item's popularity. Specifically, *PRU* is defined as:

$$PRU = -\frac{1}{|\boldsymbol{U}|}\sum_{u\in\boldsymbol{U}} SRC(pop(\boldsymbol{O_u^+}), rank(\boldsymbol{O_u^+})), \tag{19}$$

where $SRC(\cdot)$ is the Spearman Rank Correlation coefficient, and $\boldsymbol{O_u^+}$ represents the set of items interacted with by user u in the test dataset, reflecting user preferences. In particular, a smaller value of *PRU* indicates that the model more accurately captures the user's true preference, independent of item popularity.

For the user side, we employ Demographic Parity (*DP@K*) to evaluate the consistency of recommendation results among different user groups [6, 10, 50]. This metric utilizes Jensen–Shannon Divergence, $JSD(\cdot,\cdot)$, to measure the distributional distance in accuracy among different groups, specifically:

$$DP@K = JSD(ACC(G_{pop}^{user}), ACC(G_{tail}^{user})), \tag{20}$$

where $ACC(G_{pop}^{user})$ and $ACC(G_{tail}^{user})$ represent the distribution of recommendation accuracy for the popular user group and long-tail user group, respectively. A smaller value of *DP@K* indicates less bias on the user side.

5.1.3 ***Baselines.*** We implement *AURL* using three classic CF-based models as the backbone to validate its effectiveness. The three backbones are:

(1) **BPRMF** [33]: Maps user/item IDs into the representation space using matrix factorization techniques, capturing intricate relationships between users and items.
(2) **LightGCN** [18]: An advanced CF-based model based on GCNs. It captures higher-order interactions between users and items.
(3) **SimGCL** [55]: A cutting-edge Graph Contrastive Learning (GCL) method. It adds uniform noise to node representations when generating different views.

We compare our methods with various state-of-the-art models in three broad categories, as follows:

- **Re-weighting-based methods:**
  (1) **RS [32]**. The key to RS is to balance the number of interactions for high item/user and low engagement training by resampling.
  (2) **PC [65]**. This is a post-processing approach that directly modifies the prediction score by compensating based on the item/user popularity.
  (3) **Zerosum [34]**. It reduces model bias in recommendation systems by directly equalizing recommendation scores across items preferred by a user.
  (4) **r-AdjNorm [61]**. It obtains asymmetric aggregation by adjusting the gamma parameter during aggregation in the graph so that the model is more inclined to long-tail nodes.
  (5) **CPFair [27]**. It integrates fairness constraints from both consumer and producer perspectives in a joint objective framework for recommender systems based on the re-ranking approach.

Table 2. Performance of the *AURL* variant and the baseline using the BPRMF backbone with $K = 20$, where ↑ indicates that higher values are preferable and ↓ indicates that lower values are preferable.

Model	Amazon-Book				Movielens-20M				Douban-Book			
	HR↑	NDCG↑	PRU↓	DP↓	HR↑	NDCG↑	PRU↓	DP↓	HR↑	NDCG↑	PRU↓	DP↓
BPRMF	0.0318	0.0232	0.5189	0.2845	0.3291	0.2284	0.6400	0.2352	**0.1290**	0.1027	0.6692	0.2923
+RS	0.0297	0.0219	0.5178	0.2863	0.2971	0.2297	0.6242	0.1946	0.1151	0.0937	0.6065	0.2136
+PC	0.0297	0.0215	0.4944	0.2652	0.3269	0.2274	0.6208	0.2451	0.1174	0.0977	0.5840	0.1704
+Zerosum	0.0287	0.0208	0.4792	0.2373	0.3057	0.2141	0.6302	0.1772	0.1115	0.0905	0.6436	0.2029
+CPFair	0.0309	0.0225	0.5038	0.2757	0.3194	0.2215	0.6208	0.2280	0.1250	0.0995	0.6490	0.2837
+DICE	0.0310	0.0229	0.5477	0.2907	0.3204	0.2282	0.6358	0.2392	0.1213	0.1001	0.6106	0.1641
+MACR	0.0297	0.0195	0.4980	0.2704	0.3180	0.2219	0.6299	0.2019	0.1030	0.0849	0.6607	0.1845
+InvCF	0.0316	0.0230	0.5241	0.2873	0.3264	0.2261	0.6464	0.2377	0.1284	0.1016	0.6759	0.2955
+FairMI	0.0314	0.0229	0.5113	0.2803	0.3243	0.2251	0.6305	0.2318	0.1274	0.1013	0.6592	0.2880
+ *AURL*	**0.0322**	**0.0235**	**0.4063**	**0.1858**	**0.3372**	**0.2370**	**0.5168**	**0.1341**	0.1232	**0.1045**	**0.4814**	**0.1850**
Improvement(%)	**1.25%**	**1.29%**	**15.21%**	**21.70%**	**2.46%**	**3.17%**	**16.75%**	**24.32%**	**-4.59%**	**2.75%**	**17.57%**	**12.74%**

- **Decorrelation-based methods:**
  (6) **DICE [63]**. Based on the idea of decoupling, it designs a causal data framework that decomposes user interest and item popularity into two representations.
  (7) **MACR [47]**. MACR is a method based on counterfactual reasoning for eliminating the item side bias in the result.
- **Adversarial-based methods:**
  (8) **FairMI [59]**. It leverages adversarial principles to minimize mutual information between embeddings and sensitive attributes, while simultaneously maximizing it with non-sensitive information.
  (9) **InvCF [56]**. It exploits the notion that item/user representations remain unchanged despite variations in popularity semantics. By filtering out unstable or outdated popularity characteristics, InvCF learns unbiased representations.

In summary, the baselines cover a wide array of methods aimed at debiasing recommendation results. These techniques include re-sampling, re-weighting, regularization, re-ranking, causal approaches, among others. For a fair comparison, we exclude methods that necessitate additional training data, such as Autodebias [5] and DecRs [43].

5.1.4 ***Hyper-Parameter Settings.*** In all our experiments, the number of negative samples is set to 1, consistent with other studies [18, 55, 65]. We employ the Xavier initializer [15] to set up the parameters and utilize Adam [22] with a learning rate of 0.001 for optimizing all models. The representation size is fixed at 64, the batch size at 2048, and the L_2 regularization coefficient at 0.0001. For all baseline hyper-parameters, we adopt the values recommended in the respective papers and meticulously adjust them for the new dataset to optimize recommendation outcomes. We fine-tune λ_1 and λ_2 over the range $10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1$ with an appropriate step size. We run all models five times and report the mean results.

5.2 Overall performance

As demonstrated in Table 2, Table 3, and Table 4, we compare *AURL* with various debiasing methods across different backbones to assess the overall performance. To better elucidate the trade-off between recommendation accuracy and debiasing efforts, we visualize the relationships among recommendation accuracy (*NDCG*@20), item-side debiasing (*PRU*), and user-side debiasing (*DP*@20) across various models, as illustrated in Fig. 6. Based on the experimental results, we observe the following key points.

- Traditional debiasing methods, such as re-weighting, decorrelation, and adversarial approaches, have shown promise in reducing bias in recommendation results. These methods typically surpass backbone models in metrics like *PRU* and *DP*, indicating their potential to partially mitigate biases. However, these

Table 3. Performance of the *AURL* variant and the baseline using the LightGCN backbone with $K = 20$, where ↑ indicates that higher values are preferable and ↓ indicates that lower values are preferable.

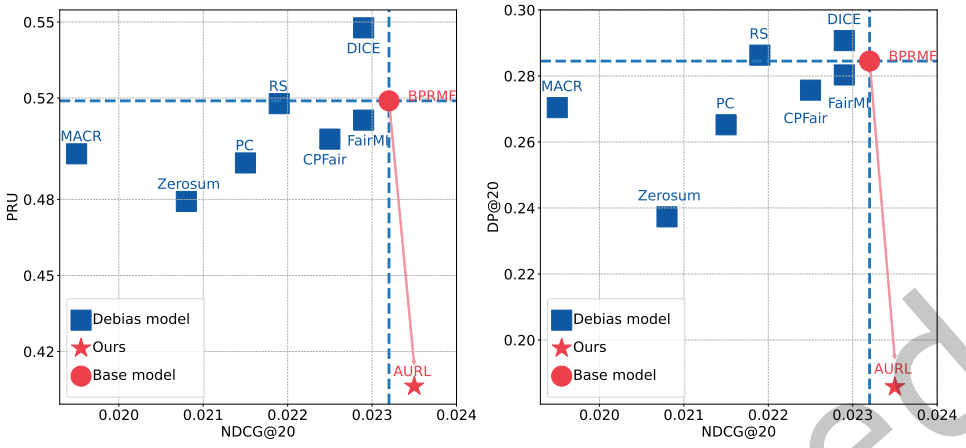
Model	Amazon-Book				Movielens-20M				Douban-Book			
	HR↑	NDCG↑	PRU↓	DP↓	HR↑	NDCG↑	PRU↓	DP↓	HR↑	NDCG↑	PRU↓	DP↓
LightGCN	0.0380	0.0282	0.4955	0.2796	0.3145	0.2172	0.3453	0.2146	0.1550	0.1282	0.6319	0.2804
+RS	0.0386	0.0280	0.4934	0.2803	0.3085	0.2130	0.3408	0.2149	0.1553	0.1285	0.5905	0.2606
+PC	0.0378	0.0282	0.4921	0.2703	0.3085	0.2130	0.3348	0.2139	0.1505	0.1230	0.5899	0.2646
+Zerosum	0.0376	0.0274	0.4911	0.2736	0.3031	0.2087	0.3342	0.2044	0.1534	0.1256	0.5719	0.2536
+r-AdjNorm	0.0387	0.0288	0.4901	0.2671	0.3214	0.2199	0.4361	0.2255	0.1664	0.1401	0.6196	0.2687
+CPFair	0.0370	0.0274	0.4950	0.2743	0.3054	0.2108	0.3360	0.2090	0.1525	0.1244	0.6137	0.2721
+DICE	0.0384	0.0284	0.4925	0.2772	0.3179	0.2164	0.3402	0.2083	0.1562	0.1285	0.6168	0.2747
+MACR	0.0356	0.0264	0.4951	0.3283	0.2905	0.1974	0.3442	0.2144	0.1392	0.1028	0.6230	0.2792
+InvCF	0.0409	0.0309	0.5028	0.2691	0.3241	0.2207	0.4846	0.2247	0.1690	0.1388	0.6260	0.2656
+FairMI	0.0391	0.0291	0.4906	0.2761	0.3244	0.2237	0.3408	0.2123	0.1597	0.1322	0.6259	0.2772
+*AURL*	**0.0458**	**0.0349**	**0.3543**	**0.2399**	**0.3495**	**0.2450**	**0.2992**	**0.1884**	**0.1831**	**0.1585**	**0.3599**	**0.2247**
Improvement(%)	**11.98%**	**12.94%**	**27.71%**	**8.99%**	**7.84%**	**11.01%**	**26.69%**	**7.82%**	**8.34%**	**13.13%**	**37.06%**	**11.40%**

Table 4. Performance of the *AURL* variant and the baseline using the SimGCL backbone with $K = 20$, where ↑ indicates that higher values are preferable and ↓ indicates that lower values are preferable.

Model	Amazon-Book				Movielens-20M				Douban-Book			
	HR↑	NDCG↑	PRU↓	DP↓	HR↑	NDCG↑	PRU↓	DP↓	HR↑	NDCG↑	PRU↓	DP↓
SimGCL	0.0537	0.0414	0.2154	0.2500	0.3603	0.2624	0.4575	0.1751	**0.1927**	**0.1677**	0.3653	0.2401
+RS	0.0517	0.0402	0.2142	0.2592	0.3583	0.2619	0.4551	0.1740	0.1925	0.1673	0.3604	0.2330
+PC	0.0537	0.0414	0.2133	0.2455	0.3596	0.2532	0.4340	0.1906	0.1862	0.1634	0.3656	0.2416
+Zerosum	0.0506	0.0403	0.2136	0.2438	0.3485	0.2508	0.4027	0.1692	0.1865	0.1570	0.3450	0.2378
+r-AdjNorm	**0.0551**	**0.0426**	0.2113	0.2736	0.3518	0.2573	0.3965	0.1629	0.1875	0.1631	0.3329	0.2675
+CPFair	0.0521	0.0400	0.2186	0.2445	0.3492	0.2551	0.4481	0.1710	0.1901	0.1640	0.3562	0.2377
+DICE	0.0526	0.0404	0.2193	0.2673	0.3603	0.2606	0.4562	0.1764	0.1921	0.1635	0.3546	0.2361
+MACR	0.0508	0.0397	0.2310	0.2858	0.3359	0.2474	0.5286	0.1770	0.1684	0.1490	0.3951	0.2792
+InvCF	0.0540	0.0411	0.2486	0.2847	0.3402	0.2495	0.4620	0.1768	0.1846	0.1539	0.3847	0.2730
+FairMI	0.0537	0.0409	0.2149	0.2482	0.3592	0.2597	0.4183	0.1667	0.1924	0.1669	0.3492	0.2342
+*AURL*	0.0541	0.0418	**0.1875**	**0.2281**	**0.3616**	**0.2632**	**0.3638**	**0.1559**	0.1884	0.1644	**0.2967**	**0.2192**
Improvement(%)	**-1.98%**	**-1.72%**	**11.27%**	**6.31%**	**0.33%**	**0.30%**	**10.76%**	**4.29%**	**-2.23%**	**-1.96%**	**10.87%**	**5.92%**

methods tend to overly focus on reducing the over-representation of popular users/items. As a result, the performance on these popular entities is deliberately decreased, leading to a notable decline in overall model performance. For example, using BPRMF as the backbone, among all traditional debiasing methods, Zerosum exhibits a relative decrease in metrics like *NDCG*@20 and *HR*@20. Nonetheless, it achieves the best performance in terms of *PRU* and *DP*@20 across various datasets. Beyond traditional techniques, we also explore state-of-the-art methods like r-adjnorm, which adjusts standardization terms in graph aggregation. We have discovered that for graph-based models, fine-tuning aggregation parameters yields the most effective debiasing results. This improvement is largely due to the ability of these parameters in GCNs to balance the influence of long-tail nodes, thus enhancing both the debiasing effect and the model's accuracy.

- *AURL* significantly outperforms existing methods in reducing biases across all datasets and backbone models. For example, compared to the best existing debiasing baselines, our method shows remarkable performance improvements on the BPRMF backbone across three real datasets, with gains of 15.21%, 21.70%, 16.75%, 24.32%, 17.57%, and 12.74% in the *PRU* and *DP*@20 metrics. Similarly, under the LightGCN backbone, improvements include 27.71%, 8.99%, 26.69%, 7.82%, 37.06%, and 11.40%. Additionally, with the SimGCL backbone, our approach enhances performance by 11.27%, 6.31%, 10.76%, 4.29%, 10.87%, and



(a) Item side (b) User side

Fig. 6. Trade-off between recommendation accuracy and debiasing efforts in the Amazon-Book dataset.

5.92%. These enhancements are the result of optimizing two critical properties in the representation space: group-alignment and global-uniformity.

- *AURL* significantly reduces biases in recommendation results while also preserving model accuracy to a notable degree, thus enhancing the model's practical utility. For instance, with BPRMF as the backbone, *AURL* achieves performance gains in *HR*@20 and *NDCG*@20 across three datasets, with improvements of 1.25%, 1.29%, 2.46%, 3.17%, -4.59%, and 2.75% respectively. With LightGCN as the backbone, these gains are even more pronounced, with increases of 11.98%, 12.94%, 7.84%, 11.01%, 8.34%, and 13.13%. These improvements demonstrate that *AURL* effectively mitigates bias without sacrificing accuracy, making it highly suitable for practical applications, as depicted in Fig. 6, where it is clear that other methods compromise accuracy to reduce bias. Additionally, *AURL* shows significantly better performance enhancements compared to other graph-based backbone models. This can be attributed to the inherent properties of graph convolution operations, where nodes assimilate information from their neighbors. Through successive convolution layers, node representations tend to converge, reducing their discriminative power between different entities and resulting in a less uniform distribution of representations in graph-based collaborative filtering methods.
- GCL-based methods like SimGCL optimize the uniformity of representations, enhancing their informativeness to a degree. Consequently, *AURL* does not significantly improve accuracy beyond what is achieved by SimGCL alone. However, the combination of SimGCL and *AURL* consistently and substantially reduces biases. Specifically, when compared with the strongest baseline methods, *AURL* achieves notable improvements in *PRU* and *DP*@20, with increases of 11.27% and 6.31% on the Amazon-Book dataset, 10.76% and 4.29% on Movielens-20M, and 10.87% and 5.92% on Douban-Book, respectively. These results underscore the significant impact of our proposed group-alignment approach in mitigating biases.

5.3 Ablation study

To demonstrate the effectiveness of the various components of *AURL*, we conducted ablation studies on the Douban-Book dataset using BPRMF+*AURL*: (1) ***AURL* w/o AL**: This variant removes the group-alignment module to examine its impact on the consistency of representation distribution among different groups. (2) ***AURL* w/o UN**: This configuration excludes the global-uniformity module to assess the influence of the global-uniformity

Table 5. Effect of different model components on BPRMF+*AURL*.

Model	Douban-book			
	HR↑	NDCG↑	PRU↓	DP↓
BPRMF	0.1290	0.1027	0.6692	0.2923
***AURL*w/o AL**	**0.1325**	0.1031	0.6231	0.2403
***AURL*w/o UN**	0.1213	0.1001	0.6116	0.1642
***AURL*w/o U**	0.1261	0.0993	0.4872	0.2623
***AURL*w/o I**	0.1236	**0.1046**	0.5571	0.1847
BPRMF+*AURL*	0.1295	0.1027	**0.4070**	**0.1558**

property on the model. (3) ***AURL* w/o U** and ***AURL* w/o I**: These versions respectively remove the regularization terms on the user and item sides, testing the model's ability to concurrently mitigate biases for both user and item sides. By comparing these variants with the complete *AURL* model, we gain insights into the specific contributions of each component and assess their impact on the overall model performance. The results of this study are presented in Table 5.

Analyzing the results of *AURL* w/o AL and *AURL* w/o UN, we observe that these modules enhance performance with respect to *PRU* by 6.89% and 8.61%, respectively. This indicates that both components effectively alleviate bias in the model. Additionally, while the group-alignment module significantly mitigates bias, it also results in a slight decrease in recommendation accuracy. In contrast, the global-uniformity module is vital for maintaining the accuracy of the recommendation model. The synergistic interaction between these two components not only ensures the accuracy of recommendations but also effectively addresses the issue of bias.

After comparing the experimental results of *AURL* w/o U and *AURL* w/o I, it is evident that addressing the distribution of either users or items separately can effectively alleviate biases in recommendation results. This demonstrates that biases are prevalent on both the user and item sides, underscoring the necessity and effectiveness of *AURL* in treating these biases within a unified framework. Specifically, when item representations are treated separately (*AURL* w/o U), there is a more significant improvement in *PRU*, with an increase of 27.19%. Conversely, addressing the user side alone (*AURL* w/o I) leads to a more substantial enhancement in the *DP*@20 metric, with a 35.89% improvement. In comparison, *AURL* w/o U shows a 20.53% improvement in *DP*@20. These results highlight that *AURL* can be specifically tailored to mitigate one-sided biases and that biases on both user and item sides are interconnected. This emphasizes the importance of simultaneously considering biases on both sides to achieve a more balanced and fair recommendation system.

5.4 Parameter sensitivity

In this section, we explore the impact of the hyperparameters λ_1 and λ_2 on the outcomes of our recommendation system. We specifically focus on how adjustments to group alignment and global uniformity influence key performance indicators such as accuracy (*NDCG*@20), item-side bias (*PRU*), and user-side bias (*DP*@20). Additionally, we examine the sensitivity of our model to the distribution of entities across the popularity spectrum, from popular to long-tail items.

5.4.1 ***Impact of the*** λ_1**.** In our objective function, the parameter λ_1 is crucial for controlling the group-alignment of representation distributions. To assess the effects of varying λ_1, we systematically modified its value from 0.001 to 1. The results of these variations are presented in the upper three subplots of Fig. 7. Our analysis reveals that, across all backbone models, increasing λ_1 leads to a consistent decline in the *PRU*@20 and *DP*@20 metrics. This pattern underscores the effectiveness of stronger group-alignment constraints in reducing biases within recommendation results. However, it is important to note that there is also a noticeable decrease in the *NDCG*@20 metric, which suggests that while enhancing group-alignment is beneficial for bias mitigation, it may adversely affect the overall recommendation accuracy.

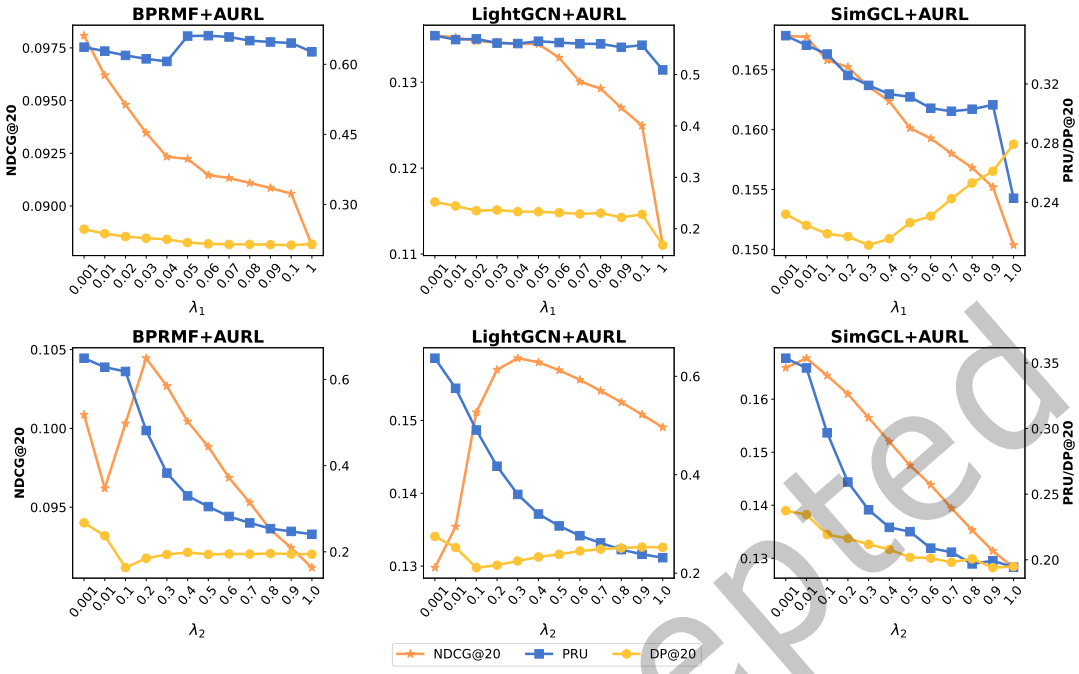
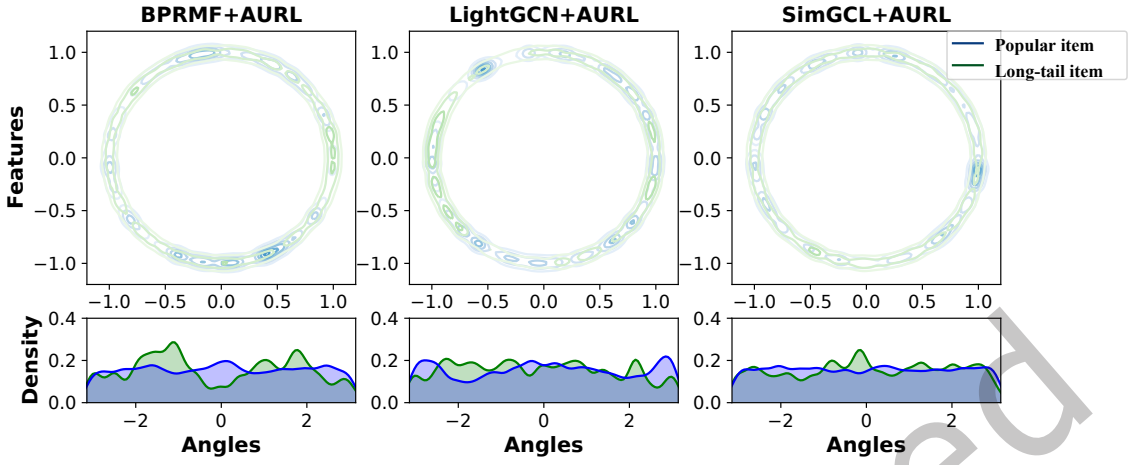


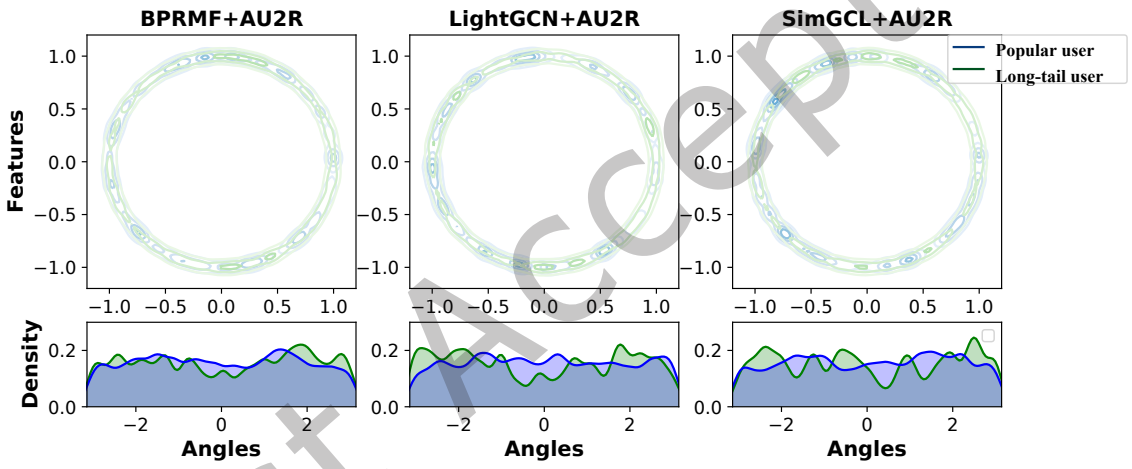
Fig. 7. Performance comparison w.r.t. different λ_1 and λ_2. The top shows the impact of λ_1 on the results, while the bottom shows the impact of λ_2 on the results. We present results for three metrics on the Douban-Book dataset: *NDCG*@20, *PRU*, and *DP*@20.

5.4.2 ***Impact of the*** λ_2***.*** In the lower three subfigures of Fig. 7, we examine the impact of the global-uniformity hyperparameter, λ_2. We were pleasantly surprised to discover that an optimal value of λ_2, particularly around 0.1, not only boosts the accuracy of recommendations but also effectively reduces biases. Specifically, both BPRMF and LightGCN models exhibit peak accuracy at this λ_2 setting. Moreover, as we increase the value of λ_2, the debiasing effect within the models becomes more pronounced. It is important to note, however, that for the SimGCL model, a higher λ_2 value results in diminished accuracy. This may be attributed to the inherent nature of SimGCL, which incorporates contrastive learning elements that already contribute to a level of global-uniformity. Despite this, our approach continues to effectively mitigate bias within the SimGCL model.

5.4.3 ***Impact of the division percentage.*** To assess the sensitivity of our model to the division ratio between popular and long-tail entities, we conducted a series of experiments varying the proportion of entities considered as 'popular'. We defined popular entities as the top 5%, 10%, 20%, and 30% of entities in the dataset. The outcomes of these experiments are summarized in Table 7. The experimental results indicate that, in most scenarios, adjusting the division percentage leads to increased accuracy and reduced bias compared to the best-performing baselines. Specifically, when the proportion of popular users or items is below 10%, our model tends to show suboptimal results in both accuracy and debiasing effectiveness. This is likely due to the insufficient representation of popular entities, which hampers the model's ability to effectively learn and align their distribution, thus diminishing overall performance. Conversely, increasing the proportion of popular entities generally improves both the accuracy and the debiasing capability of the model, particularly for user-side popularity bias. This enhancement is attributed to the more abundant interaction data available for popular users, which allows the model to more accurately capture their preferences and behavioral patterns. Our findings demonstrate that our model maintains



(a) Item representation distribution



(b) User representation distribution

Fig. 8. Visualization of *AURL* user and item representations on Douban-Book on $\mathcal{S}^1$. To improve the readability of the figure, we uniformly randomly select 500 items/users for display. We show the entities with Gaussian KDE and the angles with vMF KDE. Green and blue points represent different user or item groups.

strong performance across various division percentages, highlighting its robustness and adaptability to different entity distribution scenarios.

5.5 Visualizing the distribution of representations

To intuitively demonstrate how *AURL* mitigates recommendation biases, we employed t-SNE [39] to visualize user and item representations within the Douban-Book dataset. The distributions of representations under various backbones facilitated by *AURL* are depicted in Fig. 8 and 9.

Compared to the baseline shown in Fig. 2, the representations learned through our method display a more even and consistent distribution across the space. Notably, both popular and long-tail entity groups are uniformly dispersed and become indistinguishable from one another, enhancing the likelihood of achieving unbiased

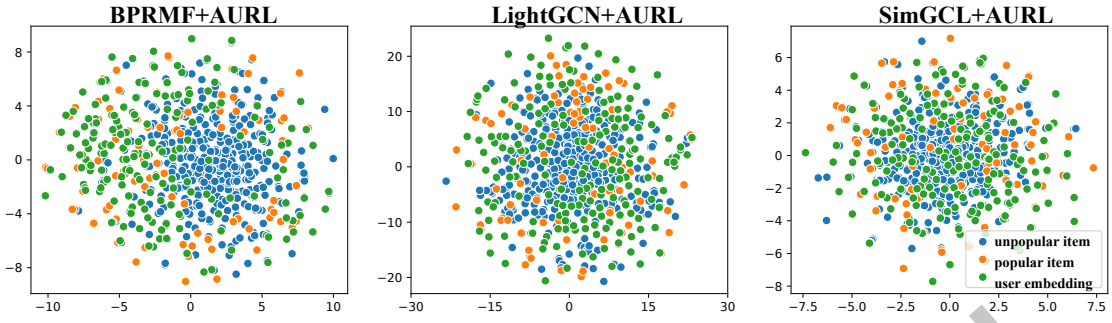


Fig. 9. Representation visualization of users and items. The green points represent users, and the blue and orange points represent popular items and long-tail items, respectively. To improve the readability of the figure, we uniformly randomly select 500 items/users to show on each subplot after performing t-SNE.

Table 7. Performance of AURL across different division percentages of popular users/items. The division percentages represent the proportion of entities considered as popular (top 5%, top 10%, top 20%, and top 30%) in the Amazon-book and Douban-book datasets.

Dataset		Amazon-book				Douban-book			
BPRMF	+Zerosum	0.0287	0.0208	0.4792	0.2373	0.1115	0.0905	0.6436	0.2029
	+InvCF	0.0316	0.0230	0.5241	0.2873	0.1284	0.1016	0.6759	0.2955
	AURL-5%	0.0300	0.0224	**0.3497**	0.2871	0.1231	0.1043	0.4800	0.2317
	AURL-10%	**0.0327**	**0.0243**	0.3915	0.2918	0.1231	0.1041	**0.4735**	0.2005
	AURL-20%	0.0322	0.0235	0.4063	**0.1858**	0.1232	**0.1045**	0.4814	0.1850
	AURL-30%	0.0325	0.0231	0.4107	0.2649	0.1229	0.1005	0.5456	0.2128
LightGCN	+Zerosum	0.0376	0.0274	0.4911	0.2736	0.1534	0.1256	0.5719	0.2536
	+InvCF	0.0409	0.0309	0.5028	0.2691	0.1690	0.1388	0.6260	0.2656
	AURL-5%	0.0459	**0.0350**	0.3661	**0.2398**	0.1830	0.1584	0.3579	0.2900
	AURL-10%	**0.0460**	0.0349	0.3633	0.2410	0.1829	0.1583	**0.3577**	0.2452
	AURL-20%	0.0458	0.0349	**0.3543**	0.2399	**0.1831**	**0.1585**	0.3599	**0.2247**
	AURL-30%	0.0451	0.0340	0.3891	0.2525	0.1807	0.1544	0.3648	0.2329
SimGCL	+Zerosum	0.0506	0.0403	0.2136	0.2438	0.1865	0.1570	0.3450	0.2378
	+InvCF	0.0540	0.0411	0.2486	0.2847	0.1846	0.1539	0.3847	0.2730
	AURL-5%	0.0491	0.0374	0.3138	0.2565	0.1845	0.1611	**0.2712**	0.2809
	AURL-10%	0.0502	0.0380	0.2903	0.2677	0.1879	0.1641	0.2828	0.2424
	AURL-20%	0.0541	0.0418	**0.1875**	**0.2281**	0.1884	0.1644	0.2967	**0.2192**
	AURL-30%	0.0537	0.0414	0.2133	0.2485	0.1893	0.1656	0.2999	0.2241

recommendation results. This uniform distribution indicates that *AURL* effectively mitigates biases on both the user and item sides as intended. Further comparisons with Fig. 3 reveal that in the representation space, users are more uniformly dispersed and are not predominantly clustered around popular items. This alteration suggests that the model recommendations are now more heavily influenced by genuine user preferences and item characteristics rather than by the popularity bias. Consequently, both popular and long-tail items are less distinguishable, leading to a more consistent distribution across the board.

Moreover, we discuss the inherent trade-off between effectiveness and debiasing. By constraining the representation distribution, *AURL* modifies the natural grouping of users and items, forcing a greater similarity

between different groups. While this approach enhances debiasing, it may occasionally reduce the effectiveness of recommendations due to altered distribution dynamics.

6 CONCLUSION AND FUTURE WORK

In this paper, we analyzed deep-rooted biases in recommender systems, focusing on group-discrepancy and global-collapse in representation distribution, which led to biased outcomes. We introduced a framework, AURL, that addressed these biases by enforcing group-alignment and global-uniformity. This approach was compatible with any CF-based model as an auxiliary task, requiring no extra parameter tuning. Our extensive evaluations across multiple domains using three datasets demonstrated that AURL outperformed existing baselines in debiasing, effectively reducing user and item biases while maintaining accuracy. For future work, we aim to identify and mitigate additional types of biases and expand our model's application to broader scenarios, thereby enhancing the fairness and accuracy of recommender systems.

ACKNOWLEDGMENTS

This work was supported in part by grants from the National Key Research and Development Program of China (Grant No. 2021ZD0111802), the National Natural Science Foundation of China (Grant Nos. U23B2031, 72188101), the China Postdoctoral Science Foundation (Grant No. 2023M741943), and the Postdoctoral Fellowship Program of CPSF (Grant No. GZC20231373).

REFERENCES

[1] Shun-ichi Amari. 1993. Backpropagation and stochastic gradient descent method. *Neurocomputing* (1993).
[2] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. *RecSys* (2018).
[3] Sergiy V Borodachov, Douglas P Hardin, and Edward B Saff. 2019. Discrete energy on rectifiable sets. *Springer* (2019).
[4] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. *SIGIR* (2019).
[5] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. *SIGIR* (2021).
[6] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *TOIS* (2020).
[7] Jiajia Chen, Jiancan Wu, Jiawei Chen, Xin Xin, Yong Li, and Xiangnan He. 2024. How graph convolutions amplify popularity bias for recommendation? *FCS* (2024).
[8] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. *SIGIR* (2017).
[9] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting Graph based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. *AAAI* (2020).
[10] Lei Chen, Le Wu, Kun Zhang, Richang Hong, Defu Lian, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Improving Recommendation Fairness via Data Augmentation. *WWW* (2023).
[11] Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. 2012. Discover breaking events with popular hashtags in twitter. *ICKM* (2012).
[12] Leyan Deng, Defu Lian, Chenwang Wu, and Enhong Chen. 2022. Graph convolution network based recommender systems: Learning guarantee and item mixture powered strategy. *NeurIPS* (2022).
[13] Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. *SIGIR* (2023).
[14] Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. 2023. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *TOIS* (2023).
[15] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. *AISTATS* (2010).
[16] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *JMLR* (2012).
[17] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *TIIS* (2015).

[18] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *SIGIR* (2020).
[19] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. *WWW* (2017).
[20] Min Hou, Le Wu, Enhong Chen, Zhi Li, Vincent W Zheng, and Qi Liu. 2019. Explainable fashion recommendation: A semantic attribute region guided approach. *IJCAI* (2019).
[21] Jack Kiefer and Jacob Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *JSTOR* (1952).
[22] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *ICLR* (2015).
[23] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* (2009).
[24] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *NeurIPS* (2017).
[25] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented fairness in recommendation. *WWW* (2021).
[26] Zhongzhou Liu, Yuan Fang, and Min Wu. 2023. Mitigating popularity bias for users and items with fairness-centric adaptive recommendation. *TOIS* (2023).
[27] Mohammadmehdi Naghiaei, Hossein A Rahmani, and Yashar Deldjoo. 2022. Cpfair: Personalized consumer and producer fairness re-ranking for recommender systems. *SIGIR* (2022).
[28] Matjaž Perc. 2014. The Matthew effect in empirical data. *J R Soc Interface* (2014).
[29] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. 2020. On minimum discrepancy estimation for deep domain adaptation. *DAVU* (2020).
[30] Weijieying Ren, Lei Wang, Kunpeng Liu, Ruocheng Guo, Lim Ee Peng, and Yanjie Fu. 2022. Mitigating popularity bias in recommendation with unbalanced interactions: A gradient perspective. *ICDM* (2022).
[31] Weijieying Ren, Pengyang Wang, Xiaolin Li, Charles E Hughes, and Yanjie Fu. 2022. Semi-supervised drifted stream learning with short lookback. *KDD* (2022).
[32] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. *WSDM* (2014).
[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. *UAI* (2009).
[34] Wondo Rhee, Sung Min Cho, and Bongwon Suh. 2022. Countering Popularity Bias by Regularizing Score Differences. *RecSys* (2022).
[35] Mary Priya Sebastian. 2023. Malayalam Natural Language Processing: Challenges in Building a Phrase-Based Statistical Machine Translation System. *TALLIP* (2023).
[36] Pengyang Shao, Le Wu, Lei Chen, Kun Zhang, and Meng Wang. 2022. FairCF: Fairness-aware collaborative filtering. *SCIS* (2022).
[37] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *NeurIPS* (2017).
[38] Ilya O. Tolstikhin, Bharath K. Sriperumbudur, and Bernhard Schölkopf. 2016. Minimax Estimation of Maximum Mean Discrepancy with Radial Kernels. *NeurIPS* (2016).
[39] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J Mach Learn Res* (2008).
[40] Thomas Viehmann. 2021. Partial Wasserstein and Maximum Mean Discrepancy distances for bridging the gap between outlier detection and drift detection. *arXiv* (2021).
[41] Chenyang Wang, Yuanqing Yu, Weizhi Ma, M. Zhang, C. Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. *KDD* (2022).
[42] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *ICML* (2020).
[43] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. *KDD* (2021).
[44] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal representation learning for out-of-distribution recommendation. *WWW* (2022).
[45] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. *SIGIR* (2019).
[46] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A survey on the fairness of recommender systems. *TOIS* (2023).
[47] Tianxin Wei, Fuli Feng, Jiawei Chen, Chufeng Shi, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2020. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. *KDD* (2020).
[48] Jiancan Wu, Xiangnan He, Xiang Wang, Qifan Wang, Weijian Chen, Jianxun Lian, and Xing Xie. 2022. Graph convolution machine for context-aware recommender system. *FCS* (2022).
[49] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2020. Self-supervised Graph Learning for Recommendation. *SIGIR* (2020).
[50] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. 2021. Learning fair representations for recommendation: A graph-based perspective. *WWW* (2021).

[51] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2023. A Survey on Accuracy-Oriented Neural Recommendation: From Collaborative Filtering to Information-Rich Recommendation. *TKDE* (2023).
[52] Yonghui Yang, Le Wu, Kun Zhang, Richang Hong, Hailin Zhou, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Hyperbolic Graph Learning for Social Recommendation. *TKDE* (2023).
[53] Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Generative-Contrastive Graph Learning for Recommendation. *SIGIR* (2023).
[54] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2023. Self-supervised learning for recommender systems: A survey. *TKDE* (2023).
[55] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Li zhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary?: Simple Graph Contrastive Learning for Recommendation. *SIGIR* (2022).
[56] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. 2023. Invariant Collaborative Filtering to Popularity Distribution Shift. *WWW* (2023).
[57] Daoan Zhang, Chenming Li, Haoquan Li, Wenjian Huang, Lingyun Huang, and Jianguo Zhang. 2023. Rethinking alignment and uniformity in unsupervised image semantic segmentation. *AAAI* (2023).
[58] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. *SIGIR* (2021).
[59] Chen Zhao, Le Wu, Pengyang Shao, Kun Zhang, Richang Hong, and Meng Wang. 2023. Fair Representation Learning for Recommendation: A Mutual Information Perspective. *AAAI* (2023).
[60] Jujia Zhao, Wenjie Wang, Xinyu Lin, Leigang Qu, Jizhi Zhang, and Tat-Seng Chua. 2023. Popularity-aware Distributionally Robust Optimization for Recommendation System. *CIKM* (2023).
[61] Minghao Zhao, Le Wu, Yile Liang, Lei Chen, Jian Zhang, Qilin Deng, Kai Wang, Xudong Shen, Tangjie Lv, and Runze Wu. 2022. Investigating Accuracy-Novelty Performance for Graph-based Collaborative Filtering. *SIGIR* (2022).
[62] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2022. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *TKDE* (2022).
[63] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling user interest and conformity for recommendation with causal embedding. *WWW* (2021).
[64] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. 2019. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. *AAAI* (2019).
[65] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. *WSDM* (2021).
[66] Zhen Zhu, Tengteng Huang, Mengde Xu, Baoguang Shi, Wenqing Cheng, and Xiang Bai. 2021. Progressive and aligned pose attention transfer for person image generation. *TPAMI* (2021).