Popularity-Aware Alignment and Contrast for Mitigating Popularity Bias

Miaomiao Cai
Hefei University of Technology
Hefei, China
cmm.hfut@gmail.com

Lei Chen
Tsinghua University
Beijing, China
chenlei.hfut@gmail.com

Yifan Wang
DCST, Tsinghua University
China, Beijing
yf-wang21@mails.tsinghua.edu.cn

Haoyue Bai
Hefei University of Technology
Hefei, China
baihaoyue621@gmail.com

Peijie Sun
DCST, Tsinghua University
Beijing, China
sun.hfut@gmail.com

Le Wu
Hefei University of Technology
Hefei, China
lewu.ustc@gmail.com

Min Zhang*
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing, China
z-m@tsinghua.edu.cn

Meng Wang*
Hefei University of Technology
Hefei, China
eric.mengwang@gmail.com

ABSTRACT

Collaborative Filtering (CF) typically suffers from the significant challenge of popularity bias due to the uneven distribution of items in real-world datasets. This bias leads to a significant accuracy gap between popular and unpopular items. It not only hinders accurate user preference understanding but also exacerbates the Matthew effect in recommendation systems. To alleviate popularity bias, existing efforts focus on emphasizing unpopular items or separating the correlation between item representations and their popularity. Despite the effectiveness, existing works still face two persistent challenges: (1) how to extract common supervision signals from popular items to improve the unpopular item representations, and (2) how to alleviate the representation separation caused by popularity bias. In this work, we conduct an empirical analysis of popularity bias and propose _Popularity-Aware Alignment and Contrast (PAAC)_ to address two challenges. Specifically, we use the common supervisory signals modeled in popular item representations and propose a novel popularity-aware supervised alignment module to learn unpopular item representations. Additionally, we suggest re-weighting the contrastive learning loss to mitigate the representation separation from a popularity-centric perspective. Finally, we validate the effectiveness and rationale of _PAAC_ in mitigating popularity bias through extensive experiments on three real-world datasets. Our code is available at https://github.com/miaomiao-cai2/KDD2024-PAAC.

*Corresponding Authors.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Collaborative Filtering, Popularity Bias, Supervised Alignment, Re-weighting, Contrastive Learning

1 INTRODUCTION

Modern recommender systems play a crucial role in mitigating information overload [3, 11, 27, 45, 48]. Collaborative filtering (CF) is widely used in personalized recommendations to help users find items of potential interest. CF-based methods primarily learn user preferences and item characteristics by aligning the representations of users and the items they interact with [19, 42]. Despite their success, CF-based methods often face popularity bias [6, 35], resulting in significant accuracy gaps between popular and unpopular items [38, 58]. Popularity bias stems from the limited supervisory signals for unpopular items, causing overfitting during training and reducing performance on the test set. It hinders the accurate understanding of user preferences, decreasing recommendation diversity [5, 16, 25]. What's even worse is that popularity bias may exacerbate the Matthew effect, where popular items become even more popular due to frequent recommendations [35, 57].

Mitigating popularity bias in recommendation systems, as described in Figure. 1, presents two primary and significant challenges. The first challenge arises from insufficient representations of unpopular items during training, leading to overfitting and poor generalization performance. The second challenge, representation
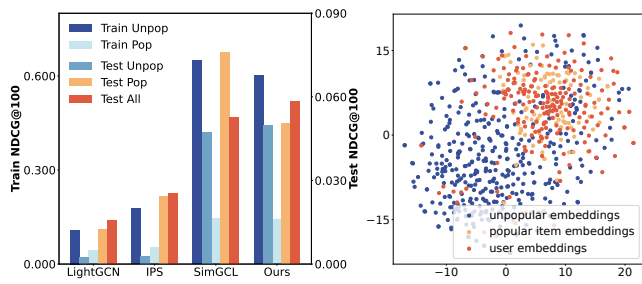
Figure 1: Popularity bias presents two challenges: (1) Overfitting caused by limited supervisory signals for unpopular items, and (2) Representation separation in item embeddings driven by popularity bias.

separation, occurs when popular and unpopular items are modeled into different semantic spaces, exacerbating bias and reducing recommendation accuracy. Next, we will explore these challenges and discuss potential solutions to mitigate popularity bias.

Due to the limited supervisory signals for unpopular items, their representations are insufficient leading to overfitting. During training, representation alignment focuses on users and the items they have interacted with [24, 30]. However, due to limited interactions, unpopular items are often modeled around a small number of users. This focused modeling can lead to overfitting due to the insufficient representations of unpopular items. As shown in the left part of Figure.1, we divide items into popular and unpopular groups based on the Pareto principle [43]. And then evaluate their performance in both training and testing sets (measured using $NDCG@100$). The results reveal that traditional methods achieve higher accuracy for unpopular items during training but significantly lower accuracy during testing, indicating clear overfitting. To address this issue, previous studies have tried to boost the training weights or prediction scores for unpopular items, such as IPS [58], MACR [38], and others [5, 9, 55]. However, as shown in Figure.1, overfitting still exists even with augmented supervisory weights for unpopular items. This may be because unpopular items still lack sufficient supervisory signals, leading to inadequate representation capability. Therefore, **how to enhance the representation modeling of unpopular items** remains a challenge.

Recent studies indicate that popularity bias causes representation separation in item embeddings [50, 54]. Specifically, the model represents popular and unpopular items in different semantic spaces according to their popularity levels. As shown in the right part of Figure.1, we train LightGCN[13] on the Yelp2018 dataset[1], randomly selected users and items, and visualize them using t-SNE [29] dimensionality reduction. The blue dots represent unpopular items, the yellow dots represent popular items, and the orange dots represent users. As seen, there is a clear distinction in the positions of unpopular and popular items in the representation space. User representations show a preference for popular items, exacerbating popularity bias. Existing methods try to alleviate representation separation by either removing the correlation between item representations and popularity [1, 38, 41] or enhancing overall consistency

[1] https://www.yelp.com/dataset

through contrastive learning [31, 50]. However, blindly removing popularity information can harm recommendation accuracy [5]. While contrastive learning methods improve recommendation performance, they often worsen representation separation by pushing positive and negative samples apart. When negative samples follow the popularity distribution [4, 50], most are popular items. Optimizing for unpopular items as positive samples pushes popular items further away, intensifying representation separation. Conversely, when negative samples follow a uniform distribution [49], most are unpopular items. Optimizing for popular items as positive samples separates them from most unpopular items, again worsening representation separation. Therefore, **how to effectively solve representation separation** is also crucial.

In this work, we conduct a analysis of popularity bias and propose _Popularity-Aware Alignment and Contrast (PAAC)_ to address two challenges. Our model _PAAC_ primarily consists of the following two modules: (1) **Supervised Alignment Module**: To enhance the representations of unpopular items with more supervision signals, we use common supervisory signals modeled in popular item representations and propose a popularity-aware supervised alignment module. Intuitively, items interacted with by the same user share similar characteristics. By leveraging similar characteristics modeled in popular item representation, we propose to align the representations of popular and unpopular items interacted with by the same user. (2) **Re-weighting Contrast Module:** To better alleviate representation separation, we propose a re-weighting contrast module from a popularity-centric perspective. Considering the influence of various popularity levels on recommendation performance as positive and negative samples, we introduce hyperparameters γ and β to control the weighting of samples with different item popularity levels. Our contributions can be summarized in three key points:

- To provide more supervisory signals for unpopular items, we leverage common characteristics modeled in popular item representations and propose a popularity-aware supervised alignment module to enhance the unpopular item representations.
- To more effectively alleviate representation separation, we propose a re-weighting contrast module from a popularity-centric perspective, re-weighting the positive and negative samples.
- Extensive experiments on three real-world datasets demonstrate the effectiveness and rationale of _PAAC_ in mitigating popularity bias.

2 PRELIMINARY

2.1 Collaborative Filtering

The core of CF-based models is to learn user preferences and item characteristics by aligning user and item representations based on their interactions [10, 30]. Based on these representations, the trained model predicts potential interactions for recommendation [20]. Specifically, let U ($|U| = M$) and I ($|I| = N$) represent the sets of users and items, respectively. In the implicit feedback setting, the observed interactions are represented by the matrix $\mathcal{R} \in {0, 1}^{M \times N}$, where $\mathcal{R}_{u,i} = 1$ indicates an interaction between user u and item i, and $\mathcal{R}_{u,i} = 0$ indicates no interaction. To better learn user preferences and item characteristics, we use LightGCN [13] as the encoder. It employs Graph Convolution Networks (GCNs) to learn high-order

collaborative signals [8, 34], mapping user/item IDs to the user representation matrix $\mathbf{Z} \in \mathbb{R}^{M \times D}$ and the item representation matrix $\mathbf{H} \in \mathbb{R}^{N \times D}$. Next, the prediction score estimates how likely user u will prefer item i based on these representations. We use the dot product [10, 24] to define the prediction score: $s(u, i) = \mathbf{z}_u^T \mathbf{h}_i$, where $s(u, i)$ denotes the prediction score for user u on item i, and $\mathbf{z}_u$ and $\mathbf{h}_i$ denote the representations of user u and item i, respectively.

To better optimize the learning of representations, many studies use the Bayesian Personalized Ranking (BPR) loss [24], a well-designed pairwise ranking objective for the recommendation. We apply this as the main loss for the recommendation task:

$$\mathcal{L}_{rec} = -\frac{1}{|\mathcal{R}|} \sum_{(u,i,j) \in O^+} ln\sigma(s(u,i) - s(u,j)), \qquad (1)$$

where $\sigma(\cdot)$ is the sigmoid function, $O^+ = \{(u,i,j)|\mathcal{R}_{u,i} = 1, \mathcal{R}_{u,j} = 0\}$ represents pairwise data, and j is a randomly sampled negative item that the user has not interacted with.

2.2 Contrastive Learning based CF

Recent studies on Contrastive Learning (CL)-based recommender systems suggest that optimizing the uniformity of items can mitigate popularity bias to some extent [15, 17, 20, 49]. Specifically, CL-based models use Information Normalized Cross Entropy (InfoNCE [22]) to minimize the distance between positive samples and maximize the distance from negative samples:

$$\mathcal{L}_{cl} = \sum_{i \in I} \log \frac{exp(h_i' h_i'' / \tau)}{\sum_{j \in I} exp(h_i' h_j'' / \tau)}, \qquad (2)$$

where i and j represent positive and negative items respectively, h' and h'' are the item representations after different data augmentations, and $\tau > 0$ is the temperature coefficient. In this work, we use noise perturbation for data augmentation, a simpler and more effective method than graph augmentation [46, 50]. Although effective, CL-based methods tend to exacerbate representation separation by increasing the distance between positive and negative samples, as illustrated in Section 1.

3 THE PROPOSED MODEL

To address the existing challenges in mitigating popularity bias, we propose _Popularity-Aware Alignment and Contrast (PAAC)_, as illustrated in Figure. 2. We leverage the common supervisory signals in popular item representations to guide the learning of unpopular representations and propose a popularity-aware supervised alignment module. Additionally, we introduce a re-weighting mechanism in the contrastive learning module to address representation separation from a popularity-centric perspective.

3.1 Supervised Alignment Module

During training, the alignment of representations typically emphasizes users and items that have interacted [24, 30], often resulting in items being closer to interacted users than non-interacted ones in the representation space. However, due to the limited interactions of unpopular items, they tend to be modeled based on a small subset of users. This narrow focus might lead to overfitting, as the representations of unpopular items may not adequately capture

their characteristics. As illustrated in Section 1, how to enhance unpopular representation modeling remains a challenge.

The difference in the number of supervisory signals is crucial in learning representations for popular and unpopular items. In particular, popular items benefit from an abundance of supervisory signals throughout the alignment process, facilitating effective learning of their representations. In contrast, unpopular items with a limited number of supervised users are more prone to overfitting. This is due to insufficient representation learning for unpopular items, highlighting the impact of supervisory signal distribution on representation quality. Intuitively, items interacted with by the same user share some similar characteristics. In this part, we leverage common supervisory signals in popular item representations and introduce a popularity-aware supervised alignment method to enhance unpopular item representations.

Specifically, we first filter items with similar characteristics based on the user's interests. For any user u, we refer to the set of items they interact with as I_u:

$$I_u = \{i | i \in I \text{ and } \mathcal{R}_{u,i} = 1 | u\}. \qquad (3)$$

Consistent with prior work [51, 58], we count the frequency $p(i)$ of each item i appearing in the training dataset as its popularity. Afterward, we group I_u based on the relative popularity of the items $p(i)$. For a clearer explanation, we divide I_u into two groups: the popular item group I_u^{pop} and the unpopular item group I_u^{unpop}:

$$\begin{aligned} I_u &= I_u^{pop} \cup I_u^{unpop}, \\ \forall i &\in I_u^{pop} \text{ and } i' \in I_u^{unpop}, \; p(i) > p(i'), \end{aligned} \qquad (4)$$

where I_u^{pop} and I_u^{unpop} are disjoint, and the popularity of each item in the popular group is greater than that of any item in the unpopular group [21]. This means that popular items receive more supervisory information than unpopular items, leading to poorer recommendation performance for unpopular items.

To address the challenge of inadequate representation learning for unpopular items, we leverage the assumption that items interacted with by the same user exhibit some similar characteristics. Specifically, we use similar supervisory signals in popular item representations to enhance the representations of unpopular items. Inspired by previous works [31, 52], we align the representations of items in I_u^{pop} and I_u^{unpop} to provide more supervisory information to unpopular items and enhance its representation, as follows:

$$\mathcal{L}_{sa} = \sum_{u \in \mathcal{U}} \frac{1}{|I_u|} \sum_{i \in I_u^{pop}, i' \in I_u^{unpop}} \|f(i) - f(i')\|^2, \qquad (5)$$

where $f(\cdot)$ is a recommendation encoder and $\mathbf{h}_i = f(i)$. By efficiently using the inherent information in the data, we provide more supervisory signals for unpopular items without introducing additional side information. This module enhances the representation of unpopular items, mitigating the overfitting issue.

3.2 Re-weighting Contrast Module

Recent studies have highlighted that popularity bias often results in a distinct representation separation of item embeddings [50, 54]. While CL-based methods aim to enhance overall uniformity by pushing negative samples, their current sampling strategies may
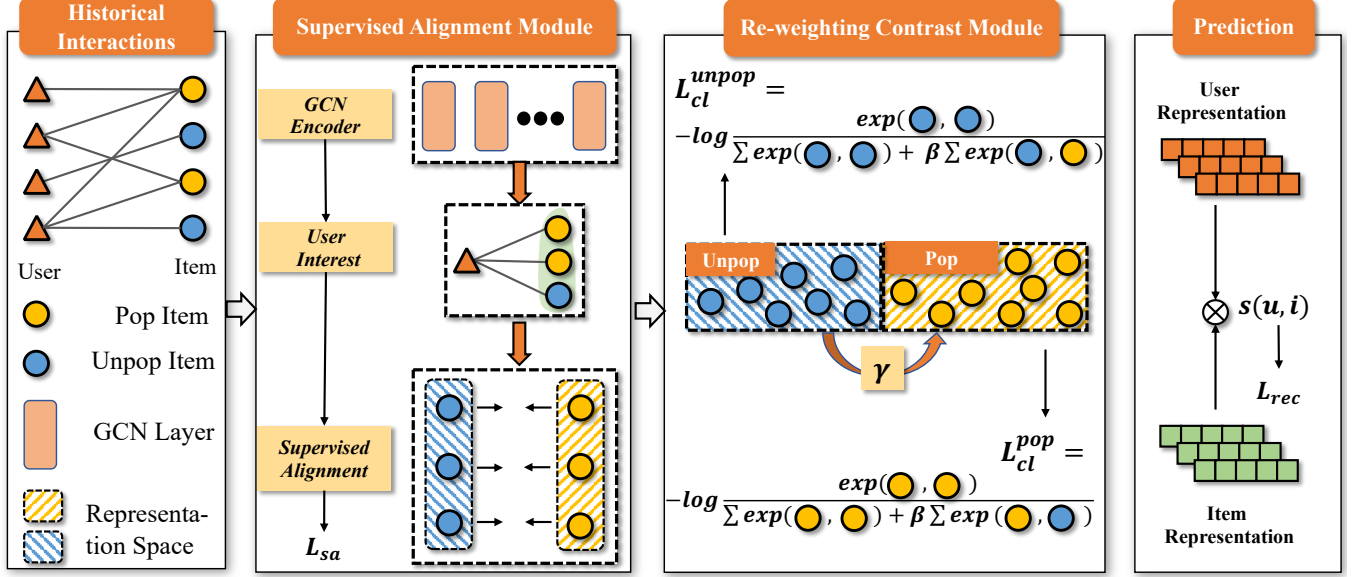
Figure 2: An Illustration of our proposed _Popularity-Aware Alignment and Contrast (PAAC)_, which consists of the Supervised Alignment Module and the Re-weighting Contrast Module. Supervised Alignment Module leverages the common supervision signal in popular representations to guide the learning of unpopular representations. Re-weighting Contrast Module address representation separation from a popularity-centric perspective.

inadvertently worsen this separation. When negative samples follow the popularity distribution [4, 50], dominated by popular items, optimizing for unpopular items as positive samples enlarges the gap between popular and unpopular items in the representation space. Conversely, when negative samples follow a uniform distribution [49], focusing on popular items separates them from the majority of unpopular ones, worsening the representation gap. Existing studies [40, 50] use the same weights for positive and negative samples in the contrastive loss function, without considering differences in item popularity. However, in real-world recommendation datasets, the impact of items varies due to dataset characteristics and interaction distributions. Neglecting this aspect could lead to suboptimal results and exacerbate representation separation.

Inspired by previous works [23, 58], we propose to identify different influences by re-weighting different popularity items. To this end, we introduce re-weighting different positive and negative samples to mitigate representation separation from a popularity-centric perspective. We incorporate this approach into contrastive learning to better optimize the consistency of representations. Specifically, we aim to reduce the risk of pushing items with varying popularity further apart. For example, when using a popular item as a positive sample, our goal is to avoid pushing unpopular items too far away. Thus, we introduce two hyperparameters to control the weights when items are considered positive and negative samples.

To ensure balanced and equitable representations of items within our model, we first propose a dynamic strategy to categorize items into popular and unpopular groups for each mini-batch. Instead of relying on a fixed global threshold, which often leads to the overrepresentation of popular items across various batches, we implement a hyperparameter x. This hyperparameter readjusts the classification of items within the current batch. By adjusting the

hyperparameter x, we maintain a balance between different item popularity levels. This enhances the model's ability to generalize across diverse item sets by accurately reflecting the popularity distribution in the current training context. Specifically, we denote the set of items within each batch as I_B. And then we divide I_B into a popular group I^{pop} and an unpopular group I^{unpop} based on their respective popularity levels, classifying the top $x\%$ of items as I^{pop}:

$$I_B = I^{pop} \cup I^{unpop},$$
$$\forall\, i \in I^{pop} \text{ and } i' \in I^{unpop},\ p(i) > p(i'), \quad (6)$$

where $I^{pop} \in I_B$ and $I^{unpop} \in I_B$ are disjoint, with I^{pop} consisting of the top $x\%$ of items in the batch. In this work, we dynamically divided items into popular and unpopular groups within each mini-batch based on their popularity, assigning the top 50% as popular items and the bottom 50% as unpopular items. This radio not only ensures equal representation of both groups in our contrastive learning but also allows items to be classified adaptively based on the batch's current composition.

After that, we use InfoNCE [22] to optimize the uniformity of item representations [31]. Unlike traditional CL-based methods, we calculate the loss for different item groups. Specifically, we introduce the hyperparameter γ to control the positive sample weights between popular and unpopular items, adapting to varying item distributions in different datasets:

$$\mathcal{L}_{cl}^{item} = \gamma * \mathcal{L}_{cl}^{pop} + (1 - \gamma) * \mathcal{L}_{cl}^{unpop}, \quad (7)$$

where $\mathcal{L}_{cl}^{pop}$ represents the contrastive loss when popular items are considered as positive samples, and $\mathcal{L}_{cl}^{unpop}$ represents the contrastive loss when unpopular items are considered as positive

samples. The value of γ ranges from 0 to 1, where $\gamma = 0$ means exclusive emphasis on the loss of popular items $\mathcal{L}_{cl}^{pop}$, and $\gamma = 1$ means exclusive emphasis on the loss of unpopular items $\mathcal{L}_{cl}^{unpop}$. By adjusting γ, we can effectively balance the impact of positive samples from both popular and unpopular items, allowing adaptability to varying item distributions in different datasets.

Following this, we fine-tune the weighting of negative samples in the contrastive learning framework using the hyperparameter β. This parameter controls how samples from different popularity groups contribute as negative samples. Specifically, we prioritize re-weighting items with popularity opposite to the positive samples, mitigating the risk of excessively pushing negative samples away and reducing representation separation. Simultaneously, this approach ensures the optimization of intra-group consistency. For instance, when dealing with popular items as positive samples, we separately calculate the impact of popular and unpopular items as negative samples. The hyperparameter β is then used to control the degree to which unpopular items are pushed away. This is formalized as follows:

$$\mathcal{L}_{cl}^{pop} = \sum_{i \in I^{pop}} \log \frac{exp(h_i' h_i'' / \tau)}{\sum\limits_{j \in I^{pop}} exp(h_i' h_j'' / \tau) + \beta \sum\limits_{j \in I^{unpop}} exp(h_i' h_j'' / \tau)}, \quad (8)$$

similarly, the contrastive loss for unpopular items is defined as:

$$\mathcal{L}_{cl}^{unpop} = \sum_{i \in I^{unpop}} \log \frac{exp(h_i' h_i'' / \tau)}{\sum\limits_{j \in I^{unpop}} exp(h_i' h_j'' / \tau) + \beta \sum\limits_{j \in I^{pop}} exp(h_i' h_j'' / \tau)}, \quad (9)$$

where the parameter β ranges from 0 to 1, controlling the negative sample weighting in the contrastive loss. When $\beta = 0$, it means that only intra-group uniformity optimization is performed. Conversely, when $\beta = 1$, it means equal treatment of both popular and unpopular items in terms of their impact on positive samples. The setting of β allows for a flexible adjustment between prioritizing intra-group uniformity and considering the impact of different popularity levels in the training. We prefer to push away items within the same group to optimize uniformity. This setup helps prevent over-optimizing the uniformity of different groups, thereby mitigating representation separation.

The final re-weighting contrastive objective is the weighted sum of the user objective and the item objective:

$$\mathcal{L}_{cl} = \frac{1}{2} \times (\mathcal{L}_{cl}^{item} + \mathcal{L}_{cl}^{user}). \quad (10)$$

In this way, we not only achieved consistency in representation but also reduced the risk of further separating items with similar characteristics into different representation spaces, thereby alleviating the issue of representation separation caused by popularity bias.

3.3 Model Optimization

To alleviate popularity bias in collaborative filtering tasks, we utilize a multi-task training strategy [40] to jointly optimize the classic recommendation loss (cf. Equation (1)), supervised alignment loss (cf. Equation (5)), and re-weighting contrast loss (cf. Equation (10)).

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_1 \mathcal{L}_{sa} + \lambda_2 \mathcal{L}_{cl} + \lambda_3 \|\Theta\|_2^2, \quad (11)$$

where Θ is the set of model parameters in $\mathcal{L}_{rec}$ as we do not introduce additional parameters, $\lambda 1$ and λ_2 are hyperparameters that control the strengths of the popularity-aware supervised alignment loss and the re-weighting contrastive learning loss respectively,

---

Algorithm 1: The Algorithm of *PAAC*

Input: user-item interactions $\mathcal{R}$, recommendation encoder $f(\cdot)$, learning rate η, hyperparameters $\lambda_1, \lambda_2, \lambda_3, \beta, \gamma$;
Output: recommendation encoder $f(\Theta)$;

1: randomly initialize recommendation encoder parameter Θ;
2: **for** $epoch = 1, 2, ..., T$ **do**
3:     **for** batch data $\mathcal{B}$ in $\mathcal{R}$ **do**
4:         calculate user and item representations;
5:         sample user u's interactions I_u in $\mathcal{B}$;
6:         divide items I_u to I_u^{pop} and I_u^{unpop} by Eqn. (4);
7:         calculate supervised alignment loss $\mathcal{L}_{sa}$ by Eqn. (5);
8:         divide items I_B to I^{pop} and I^{unpop} by Eqn. (6);
9:         calculate re-weighting contrast loss $\mathcal{L}_{cl}$ by Eqn. (10);
10:        calculate recommendation loss $\mathcal{L}_{rec}$ by Eqn. (1);
11:        calculate *PAAC* total loss $\mathcal{L}$ by Eqn. (11);
12:        update $\Theta \leftarrow \Theta - \eta * \nabla_\Theta \mathcal{L}$;
13:        **if** early stopping **then**
14:           break;
15:        **end if**
16:     **end for**
17: **end for**
18: **return** recommender encoder $f(\Theta)$.

---

and λ_3 is the L_2 regularization coefficient. After completing the model training process, we use the dot product to predict unknown preferences for recommendations. Algorithm 1 shows the detailed algorithm of *PAAC*.

4 EXPERIMENTS

In this section, we evaluate the effectiveness of *PAAC* through extensive experiments, aiming to answer the following questions:

- **RQ1**: How does *PAAC* compare to existing debiasing methods?
- **RQ2**: How do different designed components play roles in our proposed *PAAC*?
- **RQ3**: How does *PAAC* alleviate the popularity bias?
- **RQ4**: How do different hyper-parameters affect the *PAAC* recommendation performance?

4.1 Experiments Settings

*4.1.1 **Datasets.*** In our experiments, we use three widely public datasets: Amazon-book[2], Yelp2018[3], and Gowalla[4]. We retained users and items with a minimum of 10 interactions, consistent with previous works [5, 38, 50]. A detailed description can be found in Appendix A.1.

Note that the traditional dataset splitting methods fail to assess the effectiveness in mitigating popularity bias because the test sets still follow the long-tail distribution [38, 53]. In such cases, the model might perform well during testing even if it heavily relies on popularity for recommendations [6]. Hence, the conventional dataset splitting is not appropriate for evaluating whether the model suffers from popularity bias [38]. To this end, we follow previous

---
[2]https://jmcauley.ucsd.edu/data/amazon/links.html
[3]https://www.yelp.com/dataset
[4]http://snap.stanford.edu/data/loc-gowalla.html

works to extract an unbiased dataset where the item distribution in the test set follows a uniform distribution [38, 51, 54]. Specifically, we retain a fixed number of interactions for each item in the test set, amounting to approximately 10% of the entire dataset. Additionally, to avoid exposing the test distribution, we randomly selected 10% of interactions from the dataset as the validation set, and the remaining as the training set.

4.1.2 **Baselines and Evaluation Metrics.** We implement the state-of-the-art LightGCN [13] to instantiate *PAAC*, aiming to investigate how it alleviates popularity bias. We compare *PAAC* with several debiased baselines, including re-weighting-based models such as IPS [58] and γ-AdjNorm [55], decorrelation-based models like MACR [38] and InvCF [51], and contrastive learning-based models including Adap-τ [7] and SimGCL [50]. For detailed descriptions of these models, please refer to Appendix A.2.

We utilize three widely used metrics, namely *Recall@K*, *HR@K*, and *NDCG@K*, to evaluate the performance of Top-*K* recommendation. *Recall@K* and *HR@K* assess the number of target items retrieved in the recommendation results, emphasizing coverage. In contrast, *NDCG@K* evaluates the positions of target items in the ranking list, with a focus on their positions in the list. Note that we use the full ranking strategy [56], considering all non-interacted items as candidate items to avoid selection bias during the test stage [46]. We repeated each experiment five times with different random seeds and reported the average scores.

4.1.3 **Hyper-Parameter Settings.** Due to the limited space, more experimental setting details can be found in Appendix A.3.

4.2 Overall Performance (RQ1)

As shown in Table. 1, we compare our model with several baselines across three datasets. The best performance for each metric is highlighted in bold, while the second best is underlined. Our model consistently outperforms all compared methods across all metrics in every dataset.

- Our proposed model *PAAC* consistently outperforms all baselines and significantly mitigates the popularity bias. Specifically, *PAAC* enhances LightGCN, achieving improvements of 282.65%, 180.79%, and 82.89% in *NDCG@20* on the Yelp2018, Gowalla, and Amazon-Book datasets, respectively. Compared to the strongest baselines (SimGCL or Adap-τ), *PAAC* delivers better performance. The most significant improvements are observed on Yelp2018, where our model achieves an 8.70% increase in *Recall@20*, a 10.81% increase in *HR@20*, and a 30.2% increase in *NDCG@20*. This improvement can be attributed to our use of popularity-aware supervised alignment to enhance the representation of less popular items and re-weighted contrastive learning to address representation separation from a popularity-centric perspective.
- The performance improvements of *PAAC* are smaller on sparser datasets. For example, on the Gowalla dataset, the improvements in *Recall@20*, *HR@20*, and *NDCG@20* are 3.18%, 5.85%, and 5.47%, respectively. This may be because, in sparser datasets like Gowalla, even popular items are not well-represented due to lower data density. Aligning unpopular items with these poorly represented popular items can introduce noise into the

model. Therefore, the benefits of using supervisory signals for unpopular items may be reduced in very sparse environments, leading to smaller performance improvements.

- Regarding the baselines for mitigating popularity bias, the improvement of γ-Adjnorm is relatively limited compared to the backbone model (LightGCN) and even performs worse in some cases. This may be because γ-Adjnorm is specifically designed for traditional data-splitting scenarios, where the test set still follows a long-tail distribution, leading to poor generalization. IPS and MACR mitigate popularity bias by excluding item popularity information. InvCF uses invariant learning to remove popularity information at the representation level, generally performing better than IPS and MACR. This shows the importance of addressing popularity bias at the representation level. Adap-τ and SimGCL outperform the other baselines, emphasizing the necessary to improve item representation consistency for mitigating popularity bias.
- Different metrics across various datasets show varying improvements in model performance. Adapt-τ performs well on *Recall@20* and *HR@20*, while SimGCL excels in *NDCG@20*. This suggests that different debiasing methods may need distinct optimization strategies for models. Additionally, we observe varying effects of *PAAC* across different datasets, with performance improvements of 9.76%, 7.35%, and 4.83% on the Yelp2018, Amazon-Book, and Gowalla datasets, respectively. This difference could be due to the sparser nature of the Gowalla dataset. Conversely, our model can directly provide supervisory signals for unpopular items and conduct intra-group optimization, consistently maintaining optimal performance across all metrics on the three datasets.

4.3 Ablation Study (RQ2)

To better understand the effectiveness of each component in *PAAC*, we conduct ablation studies on three datasets. Table. 2 presents a comparison between *PAAC* and its variants on recommendation performance. Specifically, *PAAC*-w/o *P* refers to the variant where the re-weighting contrastive loss of popular items is removed, focusing instead on optimizing the consistency of representations for unpopular items. Similarly, *PAAC*-w/o *U* denotes the removal of the re-weighting contrastive loss for unpopular items. *PAAC*-w/o *A* refers to the variant without the popularity-aware supervised alignment loss. It's worth noting that *PAAC*-w/o *A* differs from SimGCL in that we split the contrastive loss on the item side, $\mathcal{L}_{cl}^{item}$, into two distinct losses: $\mathcal{L}_{cl}^{pop}$ and $\mathcal{L}_{cl}^{unpop}$. This approach allows us to separately address the consistency of popular and unpopular item representations, thereby providing a more detailed analysis of the impact of each component on the overall performance.

From Table. 2, we observe that *PAAC*-w/o *A* outperforms SimGCL in most cases. This validates that re-weighting the importance of popular and unpopular items can effectively improve the model's performance in alleviating popularity bias. It also demonstrates the effectiveness of using supervision signals from popular items to enhance the representations of unpopular items, providing more opportunities for future research on mitigating popularity bias. Moreover, compared with *PAAC*-w/o *U*, *PAAC*-w/o *P* results in much worse performance. This confirms the importance of re-weighting

Table 1: Performance comparison on three public datasets with $K = 20$**. The best performance is indicated in bold, while the second-best performance is underlined. The superscripts** $*$ **indicate** $p \le 0.05$ **for the paired t-test of** *PAAC* **vs. the best baseline (the relative improvements are denoted as Imp.).**

Model	Yelp2018			Gowalla			Amazon-book		
	Recall@20	*HR@20*	*NDCG@20*	*Recall@20*	*HR@20*	*NDCG@20*	*Recall@20*	*HR@20*	*NDCG@20*
MF	0.0050	0.0109	0.0093	0.0343	0.0422	0.0280	0.0370	0.0388	0.0270
LightGCN	0.0048	0.0111	0.0098	0.0380	0.0468	0.0302	0.0421	0.0439	0.0304
IPS	0.0104	0.0183	0.0158	0.0562	0.0670	0.0444	0.0488	0.0510	0.0365
MACR	0.0402	0.0312	0.0265	0.0908	0.1086	0.0600	0.0515	0.0609	0.0487
γ-**Adjnorm**	0.0053	0.0088	0.0080	0.0328	0.0409	0.0267	0.0422	0.0450	0.0264
InvCF	0.0444	0.0344	0.0291	0.1001	0.1202	0.0662	0.0562	0.0665	0.0515
Adap-τ	<u>0.0450</u>	0.0497	0.0341	0.1182	<u>0.1248</u>	0.0794	<u>0.0641</u>	<u>0.0678</u>	0.0511
SimGCL	0.0449	<u>0.0518</u>	<u>0.0345</u>	<u>0.1194</u>	0.1228	<u>0.0804</u>	0.0628	0.0648	<u>0.0525</u>
PAAC	**0.0494***	**0.0574***	**0.0375***	**0.1232***	**0.1321***	**0.0848***	**0.0701***	**0.0724***	**0.0556***
Imp.	**+9.78 %**	**+10.81%**	**+8.70%**	**+3.18%**	**+5.85%**	**+5.47%**	**+9.36%**	**+6.78%**	**5.90%**

Table 2: Ablation study of *PAAC,* **highlighting the best-performing model on each dataset and metrics in bold. Specifically,** *PAAC-***w/o** P **removes the re-weighting contrastive loss of popular items,** *PAAC-***w/o** U **eliminates the re-weighting contrastive loss of unpopular items, and** *PAAC-***w/o** A **omits the popularity-aware supervised alignment loss.**

Model	Yelp2018			Gowalla			Amazon-book		
	Recall@20	*HR@20*	*NDCG@20*	*Recall@20*	*HR@20*	*NDCG@20*	*Recall@20*	*HR@20*	*NDCG@20*
SimGCL	0.0449	0.0518	0.0345	0.1194	0.1228	0.0804	0.0628	0.0648	0.0525
*PAAC-***w/o** P	0.0443	0.0536	0.0340	0.1098	0.1191	0.0750	0.0616	0.0639	0.0458
*PAAC-***w/0** U	0.0462	0.0545	0.0358	0.1120	0.1179	0.0752	0.0594	0.0617	0.0464
*PAAC-***w/0** A	0.0466	0.0547	0.0360	0.1195	0.1260	0.0815	0.0687	0.0711	0.0536
PAAC	**0.0494***	**0.0574***	**0.0375***	**0.1232***	**0.1321***	**0.0848***	**0.0701***	**0.0724***	**0.0556***

popular items in contrastive learning for mitigating popularity bias. Finally, *PAAC* consistently outperforms the three variants, demonstrating the effectiveness of combining supervised alignment and re-weighting contrastive learning. Based on the above analysis, we conclude that leveraging supervisory signals from popular item representations can better optimize representations for unpopular items, and re-weighting contrastive learning allows the model to focus on more informative or critical samples, thereby improving overall performance. All the proposed modules significantly contribute to alleviating popularity bias.

4.4 Debias Ability (RQ3)

To further verify the effectiveness of *PAAC* in alleviating popularity bias, we conduct a comprehensive analysis focusing on the recommendation performance across different popularity item groups. Specifically, 20% of the most popular items are labeled 'Popular', and the rest are labeled 'Unpopular'. As shown in Figure. 3, we compare the performance of *PAAC* with LightGCN, IPS, MACR, and SimGCL using the *NDCG@20* metric across different popularity groups. We use Δ to denote the accuracy gap between the two groups. From Figure. 3, we draw the following conclusions:

- Our proposed *PAAC* significantly enhances the recommendation performance for unpopular items. Specifically, we observe an improvement of 8.94% and 7.30% in *NDCG@20* relative to SimGCL on the Gowalla and Yelp2018 datasets, respectively. This improvement is due to the popularity-aware supervised

alignment method, which uses supervisory signals from popular items to improve the representations of unpopular items.

- *PAAC* has successfully narrowed the accuracy gap between different item groups. Specifically, *PAAC* achieved the smallest gap, reducing the *NDCG@20* accuracy gap by 34.18% and 87.50% on the Gowalla and Yelp2018 datasets, respectively. This indicates that our method treats items from different groups fairly, effectively alleviating the impact of popularity bias. This success can be attributed to our re-weighted contrast module, which addresses representation separation from a popularity-centric perspective, resulting in more consistent recommendation results across different groups.

- Improving the performance of unpopular items is crucial for enhancing overall model performance. Specially, on the Yelp2018 dataset, *PAAC* shows reduced accuracy in recommending popular items, with a notable decrease of 20.14% compared to SimGCL. However, despite this decrease, the overall recommendation accuracy surpasses that of SimGCL by 11.94%, primarily due to a 6.81% improvement in recommending unpopular items. This improvement highlights the importance of better recommendations for unpopular items and emphasizes their crucial role in enhancing overall model performance.

Due to space limitations, we demonstrate the debiasing capabilities of our model from more dimensions in Appendix A.4, such as conventional test sets and representation separation analysis.
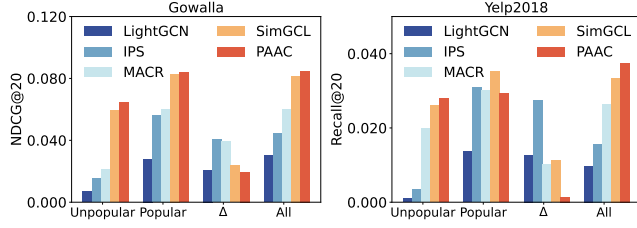
Figure 3: Performance comparison over different item popularity groups. In particular, Δ indicates the accuracy gap between different groups.
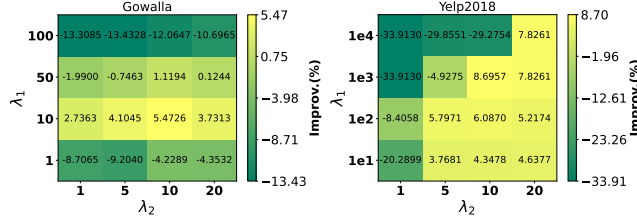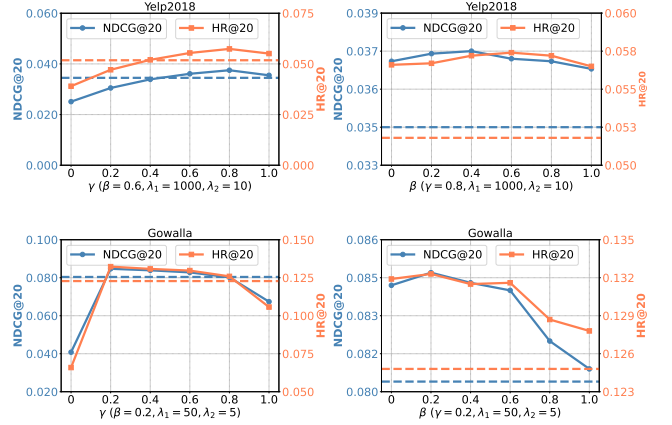

Figure 4: Performance comparison w.r.t. λ_1 and λ_2 on the Yelp2018 and Gowalla dataset in $NDCG@20$. The values indicate the percentage improvement relative to the best baseline.

4.5 Hyperparameter Sensitivities (RQ4)

In this section, we analyze the impact of hyperparameters in *PAAC*. Firstly, we investigate the influence of λ_1 and λ_2, which respectively control the impact of the popularity-aware supervised alignment and re-weighting contrast loss. Additionally, in the re-weighting contrastive loss, we introduce two hyperparameters, γ and β, to control the re-weighting of different popularity items as positive and negative samples. Finally, we explore the impact of the grouping ratio x on the model's performance.

4.5.1 **Effect of λ_1 and λ_2.** As formulated in Eq. (11), λ_1 controls the extent of providing additional supervisory signals for unpopular items, while λ_2 controls the extent of optimizing representation consistency. Figure. 4 illustrates how the relative performance to the best baseline $NDCG@20$ varies with λ_1 and λ_2 on the Yelp2018 and Gowalla datasets. **Horizontally**, with the increase in λ_2, the performance initially increases and then decreases. This indicates that appropriate re-weighting contrastive loss effectively enhances the consistency of representation distributions, mitigating popularity bias. However, overly strong contrastive loss may lead the model to neglect recommendation accuracy. **Vertically**, as λ_1 increases, the performance also initially increases and then decreases. This suggests that suitable alignment can provide beneficial supervisory signals for unpopular items, while too strong an alignment may introduce more noise from popular items to unpopular ones, thereby impacting recommendation performance.

4.5.2 **Effect of re-weighting coefficient γ and β.** To mitigate representation separation due to imbalanced positive and negative sampling, we introduce two hyperparameters into the contrastive loss. Specifically, γ controls the weight difference between positive

Figure 5: Performance comparison w.r.t. different γ and β. The top shows the $NDCG@20$ and $HR@20$ results on Yelp2018 and the bottom shows the results on Gowalla. The horizontal line represents the best results already achieved in the baseline.

Table 3: Performance comparison across varying popular item ratios x on metrics.

Ratio	Yelp2018			Gowalla		
	$Recall@20$	$HR@20$	$NDCG@20$	$Recall@20$	$HR@20$	$NDCG@20$
20%	0.0467	0.0555	0.0361	0.1232	0.1319	0.0845
40%	0.0505	0.0581	0.0378	0.1239	0.1325	0.0848
50%	0.0494	0.0574	0.0375	0.1232	0.1321	0.0848
60%	0.0492	0.0569	0.0370	0.1225	0.1314	0.0843
80%	0.0467	0.0545	0.0350	0.1176	0.1270	0.0818

samples from popular and unpopular items, while β controls the influence of different popularity items as negative samples.

In our experiments, while keeping other hyperparameters constant, we search γ and β within the range {0, 0.2, 0.4, 0.6, 0.8, 1}. Figure. 5 illustrates how performance changes when varying γ and β on two datasets, with horizontal lines representing the best baseline. As γ and β increase, performance initially improves and then declines. The optimal hyperparameters for the Yelp2018 and Gowalla datasets are $\gamma = 0.8$, $\beta = 0.6$ and $\gamma = 0.2$, $\beta = 0.2$, respectively. This may be attributed to the characteristics of the datasets. The Yelp2018 dataset, with a higher average interaction frequency per item, benefits more from a higher weight γ for popular items as positive samples. Conversely, the Gowalla dataset, being relatively sparse, prefers a smaller γ. This indicates the importance of considering dataset characteristics when adjusting the contributions of popular and unpopular items to the model.

Notably, γ and β are not highly sensitive within the range $[0, 1]$, performing well across a broad spectrum. Figure. 5 shows that performance exceeds the baseline regardless of β values when other parameters are optimal. Additionally, γ values from $[0.4, 1.0]$ on the Yelp2018 dataset and $[0.2, 0.8]$ on the Gowalla dataset surpass the baseline, indicating less need for precise tuning. Thus, γ and β achieve optimal performance without meticulous adjustments, focusing on weight coefficients to maintain model efficacy.

4.5.3 **Effect of grouping ratio x.** To investigate the impact of different grouping ratios on recommendation performance, we developed a flexible classification method for items within each minibatch based on their popularity. Instead of adopting a fixed global

threshold, which tends to overrepresent popular items in some mini-batches, our approach dynamically divides items in each mini-batch into popular and unpopular categories. Specifically, the top $x\%$ of items are classified as popular and the remaining $(100 - x)\%$ as unpopular, with x varying. This strategy prevents the overrepresentation typical in fixed distribution models, which could skew the learning process and degrade performance. To quantify the effects of these varying ratios, we examined various division ratios for popular items, including 20%, 40%, 60%, and 80%, as shown in Table. 3. The preliminary results indicate that both extremely low and high ratios negatively affect model performance, thereby underscoring the superiority of our dynamic data partitioning approach. Moreover, within the 40%-60% range, our model's performance remained consistently robust, further validating the effectiveness of *PAAC*.

5 RELATED WORK

5.1 Popularity Bias in Recommendation

Popularity bias is a common issue in recommender systems where unpopular items in the training dataset are rarely recommended [32, 38]. Many methods [2, 6, 35–37] have been proposed to analyze and reduce performance differences between popular and unpopular items. These methods can be broadly categorized into three types.

• **Re-weighting-based methods** aim to increase the training weight or scores for unpopular items, redirecting focus away from popular items during training or prediction [5, 55, 58]. For instance, IPS [58] adds compensation to unpopular items and adjusts the prediction of the user-item preference matrix, resulting in higher preference scores and improving rankings for unpopular items. γ-AdjNorm [55] enhances the focus on unpopular items by controlling the normalization strength during the neighborhood aggregation process in GCN-based models.

• **Decorrelation-based methods** aim to effectively remove the correlations between item representations (or prediction scores) and popularity[1, 26, 33, 38, 41, 53]. For instance, MACR [38] uses counterfactual reasoning to eliminate the direct impact of popularity on item outcomes. In contrast, InvCF [51] operates on the principle that item representations remain invariant to changes in popularity semantics, filtering out unstable or outdated popularity characteristics to learn unbiased representations.

• **Contrastive-learning-based methods** aim to achieve overall uniformity in item representations using InfoNCE [15, 44], preserving more inherent characteristics of items to mitigate popularity bias [31, 50]. This approach has been demonstrated as a state-of-the-art method for alleviating popularity bias. It employs data augmentation techniques such as graph augmentation or feature augmentation to generate different views, maximizing positive pair consistency and minimizing negative pair consistency to promote more uniform representations [49]. Specifically, Adap-τ[7] adjusts user/item embeddings to specific values, while SimGCL[50] integrates InfoNCE loss to enhance representation uniformity and alleviate popularity bias.

5.2 Representation Learning for CF

Representation learning is crucial in recommendation systems, especially in modern collaborative filtering (CF) techniques. It creates personalized embeddings that capture user preferences and item

characteristics [15, 24, 39, 42, 47]. The quality of these representations critically determines a recommender system's effectiveness by precisely capturing the interplay between user interests and item features [14, 30, 50]. Recent studies emphasize two fundamental principles in representation learning: alignment and uniformity [28, 30, 31]. The alignment principle ensures that embeddings of similar or related items (or users) are closely clustered together, improving the system's ability to recommend items that align with a user's interests [31]. This principle is crucial when accurately reflecting user preferences through corresponding item characteristics [30]. Conversely, the uniformity principle ensures a balanced distribution of all embeddings across the representation space [40, 50]. This approach prevents the over-concentration of embeddings in specific areas, enhancing recommendation diversity and improving generalization to unseen data [50].

In this work, we focus on aligning the representations of popular and unpopular items interacted with by the same user and re-weighting uniformity to mitigate representation separation. Our model *PAAC* uniquely addresses popularity bias by combining group alignment and contrastive learning, a first in the field. Unlike previous works that align positive user-item pairs or contrastive pairs, *PAAC* directly aligns popular and unpopular items, leveraging the rich information of popular items to enhance the representations of unpopular items and reduce overfitting. Additionally, we introduce targeted re-weighting from a popularity-centric perspective to achieve a more balanced representation.

6 CONCLUSION

In this work, we analyzed popularity bias and proposed *PAAC* to mitigate popularity bias. We assumed that items interacted with by the same user share similar characteristics and used this observation to align representations of both popular and unpopular items through a popularity-aware supervised alignment approach. This provided more supervisory information for unpopular items. Note that our hypothesis of aligning and grouping items based on user-specific preferences offers a novel alignment perspective. Additionally, we addressed the issue of representation separation in current CL-based models by introducing two hyper-parameters to control the weights of items with different popularity levels as positive and negative samples. This approach optimized representation consistency and effectively alleviated separation. Our method, *PAAC*, was validated on three public datasets, proving its rationale and effectiveness.

In the future, we will explore deeper alignment and contrast adjustments tailored to specific tasks to further mitigate popularity bias. We aim to investigate the synergies between alignment and contrast and extend our approach to address other biases in recommendation systems.

REFERENCES

[1] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. *RecSys* (2018), 104–112.

[2] Miaomiao Cai, Min Hou, Lei Chen, Le Wu, Haoyue Bai, Yong Li, and Meng Wang. 2024. Mitigating Recommendation Biases via Group-Alignment and Global-Uniformity in Representation Learning. *TIST* (2024).

[3] 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), 225–234.

[4] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *TOIS* 38, 2 (2020), 1–28.

[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), 21–30.

[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* 41, 3 (2020), 1–39.

[7] Jiawei Chen, Junkang Wu, Jiancan Wu, Xuezhi Cao, Sheng Zhou, and Xiangnan He. 2023. Adap-τ: Adaptively modulating embedding magnitude for recommendation. *WWW* (2023), 1085–1096.

[8] 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* 34, 01 (2020), 27–34.

[9] 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), 1012–1020.

[10] Lei Chen, Le Wu, Kun Zhang, Richang Hong, and Meng Wang. 2021. Set2setRank: Collaborative set to set ranking for implicit feedback based recommendation. *SIGIR* (2021), 585–594.

[11] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. *RecSys* (2016), 191–198.

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *NeurIPS* 29 (2016).

[13] 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), 639–648.

[14] Zhuangzhuang He, Yifan Wang, Yonghui Yang, Peijie Sun, Le Wu, Haoyue Bai, Jinqi Gong, Richang Hong, and Min Zhang. 2024. Double Correction Framework for Denoising Recommendation. *arXiv preprint arXiv:2405.11272* (2024).

[15] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2020), 2.

[16] Meng Jiang, Keqin Bao, Jizhi Zhang, Wenjie Wang, Zhengyi Yang, Fuli Feng, and Xiangnan He. 2024. Item-side Fairness of Large Language Model-based Recommendation System. *WWW* (2024), 4717–4726.

[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *NeurIPS* 33 (2020), 18661–18673.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

[20] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. *WWW* (2022), 2320–2329.

[21] Zhongzhou Liu, Yuan Fang, and Min Wu. 2023. Mitigating popularity bias for users and items with fairness-centric adaptive recommendation. *TOIS* 41, 3 (2023), 1–27.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[23] Seongmin Park, Mincheol Yoon, Jae-woong Lee, Hogun Park, and Jongwuk Lee. 2023. Toward a Better Understanding of Loss Functions for Collaborative Filtering. *CIKM* (2023), 2034–2043.

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *arXiv preprint arXiv:1205.2618* (2012).

[25] Pengyang Shao, Le Wu, Lei Chen, Kun Zhang, and Meng Wang. 2022. FairCF: Fairness-aware collaborative filtering. *SCIS* 65, 12 (2022), 222102.

[26] Pengyang Shao, Le Wu, Kun Zhang, Defu Lian, Richang Hong, Yong Li, and Meng Wang. 2024. Average User-Side Counterfactual Fairness for Collaborative Filtering. *TOIS* 42, 5 (2024).

[27] Peijie Sun, Yifan Wang, Min Zhang, Chuhan Wu, Yan Fang, Hong Zhu, Yuan Fang, and Meng Wang. 2024. Collaborative-Enhanced Prediction of Spending on Newly Downloaded Mobile Games under Consumption Uncertainty. *WWW2024, Industry Track* (2024), 10–19.

[28] Peijie Sun, Le Wu, Kun Zhang, Xiangzhi Chen, and Meng Wang. 2024. Neighborhood-Enhanced Supervised Contrastive Learning for Collaborative Filtering. *TKDE* 36, 5 (2024), 2069–2081.

[29] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* 9, 11 (2008).

[30] 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), 1816–1825.

[31] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *ICML* (2020), 9929–9939.

[32] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. *KDD* (2021), 1717–1725.

[33] 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), 3562–3571.

[34] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. *SIGIR* (2019), 165–174.

[35] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A survey on the fairness of recommender systems. *TOIS* 41, 3 (2023), 1–43.

[36] Yifan Wang, Peijie Sun, Weizhi Ma, Min Zhang, Yuan Zhang, Peng Jiang, and Shaoping Ma. 2024. Intersectional Two-sided Fairness in Recommendation. *WWW* (2024), 3609–3620.

[37] Yifan Wang, Peijie Sun, Min Zhang, Qinglin Jia, Jingjie Li, and Shaoping Ma. 2023. Unbiased Delayed Feedback Label Correction for Conversion Rate Prediction. *KDD* (2023), 2456–2466.

[38] 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), 1791–1800.

[39] Chenwang Wu, Xiting Wang, Defu Lian, Xing Xie, and Enhong Chen. 2023. A causality inspired framework for model interpretation. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023), 2731–2741.

[40] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2020. Self-supervised Graph Learning for Recommendation. *SIGIR* (2020), 726–735.

[41] 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), 2198–2208.

[42] 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* 35, 5 (2023), 4425–4445.

[43] Menglin Yang, Zhihao Li, Min Zhou, Jiahong Liu, and Irwin King. 2022. Hicf: Hyperbolic informative collaborative filtering. *KDD* (2022), 2212–2221.

[44] Yonghui Yang, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2021. Enhanced graph learning for collaborative filtering via mutual information maximization. *SIGIR* (2021), 71–80.

[45] Yonghui Yang, Le Wu, Zihan Wang, Zhuangzhuang He, Richang Hong, and Meng Wang. 2024. Graph Bottlenecked Social Recommendation. *Arxiv* (2024).

[46] 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), 1117–1126.

[47] Ziyi Ye, Xiaohui Xie, Yiqun Liu, Zhihong Wang, Xuesong Chen, Min Zhang, and Shaoping Ma. 2022. Towards a better understanding of human reading comprehension with brain signals. *WWW* (2022), 380–391.

[48] Ziyi Ye, Xiaohui Xie, Yiqun Liu, Zhihong Wang, Xuancheng Li, Jiaji Li, Xuesong Chen, Min Zhang, and Shaoping Ma. 2022. Why Don't You Click: Understanding Non-Click Results in Web Search with Brain Signals. *SGIR* (2022), 633–645.

[49] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2023. Self-supervised learning for recommender systems: A survey. *TKDE* 36, 1 (2023), 335–355.

[50] 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), 1240–1251.

[51] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. 2023. Invariant Collaborative Filtering to Popularity Distribution Shift. (2023).

[52] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Re. 2022. Correct-N-Contrast: a Contrastive Approach for Improving Robustness to Spurious Correlations. *arXiv preprint arXiv:2203.01517* (2022).

[53] 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), 11–20.

[54] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, Irwin King, et al. 2023. Mitigating the Popularity Bias of Graph Collaborative Filtering: A Dimensional Collapse Perspective. *NeurIPS* 36 (2023), 67533–67550.

[55] 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), 50–59.

[56] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting alternative experimental settings for evaluating top-n item recommendation algorithms. *CIKM* (2020), 2329–2332.

[57] Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity Bias in Dynamic Recommendation. *KDD* (2021), 2439–2449.

[58] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. *WSDM* (2021), 85–93.

A APPENDIX

A.1 Datasets

We conducted experiments on three public datasets, with the processed dataset statistics summarized in Table. 4. Additionally, to analyze the imbalance in item distribution, we recorded the number of interactions for the most and least popular items and calculated the average number of interactions per item. We also used the Gini coefficient to reflect the disparity in item popularity distribution [23]. The results are presented in Table. 5.

Amazon-Book. Amazon is a frequently utilized dataset for item recommendations. From this collection, we specifically selected the Amazon-Book dataset.

Yelp2018. This dataset is sourced from the 2018 edition of the Yelp Challenge, where local businesses such as restaurants and bars are considered items.

Gowalla. This is a check-in dataset from Gowalla, that contains user location data shared through check-ins.

Table 4: The statistics of three datasets.

Datasets	#Users	#Itmes	#Interactions	Density
Amazon-Book	52,643	91,599	2,984,108	0.0619%
Yelp2018	31,668	38,048	1,561,406	0.1300%
Gowalla	29,858	40,981	1,027,370	0.0840%

Table 5: The analysis of item popularity distribution.

Datasets	#Max	#Min	#Average	GINI
Amazon-Book	1902	5	27.58	0.55
Yelp2018	744	3	18.28	0.58
Gowalla	2305	5	20.07	0.55

A.2 Baselines

We compare *PAAC* with several debiased baselines, including reweighting-based models such as IPS and γ-AdjNorm, decorrelation-based models like MACR and InvCF, and contrastive learning-based models including Adap-τ and SimGCL.

- **BPRMF [24]** is a traditional CF-based model that maps user and item IDs into a representation space via matrix factorization. It optimizes using Bayesian Personalized Ranking (BPR) loss;
- **LightGCN [13]** is a state-of-the-art CF-based model that effectively captures high-order collaborative signals between users and items by linearly propagating embeddings on the user-item interaction graph through multiple layers;

- **IPS [58]** adjusts the weight of each user-item interaction according to item popularity, aiming to mitigate popularity bias;
- **MACR [38]** estimates and eliminates the direct influence of item popularity on prediction scores, using counterfactual inference to mitigate popularity bias;
- **γ-AdjNorm [55]** modifies the hyper-parameter γ to preferentially treat unpopular items in graph aggregation, resulting in non-symmetric aggregation;
- **InvCF [51]** learns unbiased preference representations that remain stable regardless of item popularity, simultaneously removing unstable and outdated characteristics;
- **Adap-τ [7]** dynamically standardizes user and item embeddings to specific values to reduce popularity bias;
- **SimGCL [50]** aims to achieve a more uniform distribution of representations by incorporating the InfoNCE loss, which helps in mitigating popularity bias.

A.3 Hyper-Parameter Settings

We initialize parameters using the Xavier initializer [12] and use the Adam optimizer [18] with a learning rate of 0.001. The embedding size is fixed at 64. For all datasets, we set the batch size is 2048, and the L_2 regularization coefficient λ_3 is 0.0001. For our model, we fine-tune the popularity-aware supervised alignment coefficient λ_1 in {1, 5, 10, 50, 100, 300, 400, 500, 100}, the popularity-aware contrastive regularization coefficient λ_2 in {0.1, 1, 5, 10, 20}, and the re-weighting hyperparameters γ and β in {0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0}. Additionally, we set the grouping ratio x=50. This means we dynamically divided items into popular and unpopular groups within each mini-batch based on their popularity, assigning the top 50% as popular items and the bottom 50% as unpopular items. This approach ensures equal representation of both groups in our contrastive learning and allows items to be adaptively classified based on the batch's current composition. Moreover, we carefully search the hyper-parameters for all baselines to ensure fair comparisons.

A.4 Debias Ability

A.4.1 Conventional Test. As mentioned in Section 4.1.1, traditional evaluation methods fail to accurately measure a model's ability to mitigate popularity bias, as test sets often exhibit a long-tail distribution. This can misleadingly suggest high performance for models that favor popular items. To address this significant issue, we utilize an unbiased dataset for evaluation, ensuring a uniform item distribution in the test set, following established precedents. Despite this, traditional performance metrics remain essential for effectively assessing model performance. Therefore, we also examine *PAAC*'s results under conventional experimental settings. These results, detailed in Table. 6, demonstrate that our method competes closely with the best baselines, affirming its efficacy under standard evaluation conditions.

A.4.2 Impact on Embedding Separation. To evaluate the impact of *PAAC* on embedding separation, we conduct detailed quantitative and qualitative analyses from various perspectives.

Quantitative Analysis. Following the Pareto principle [43], items are categorized into 'Popular' and 'Unpopular' groups. To quantify the differences in the distribution of embeddings between

Table 6: Performance comparison on conventional test.

Ratio	Yelp2018			Gowalla		
	Recall@20	*HR@20*	*NDCG@20*	*Recall@20*	*HR@20*	*NDCG@20*
LightGCN	0.0591	0.0614	0.0483	0.1637	0.1672	0.1381
Adapt-τ	0.0724	0.0753	0.0603	0.1889	0.1930	0.1584
SimGCL	0.0720	0.0748	0.0594	0.1817	0.1858	0.1526
PAAC	0.0722	0.0755	0.0602	0.1890	0.1928	0.1585

Table 7: Comparative Analysis of Maximum Mean Discrepancy (MMD) and Cosine Similarity (CS) Metrics. This table illustrates the effectiveness of the PAAC method in reducing distribution disparities between popular and unpopular item groups. The lower MMD and CS values indicate improved embedding separation, demonstrating the efficacy of the PAAC method.

Metrics	MMD↓			CS↓		
Model	LightGCN	SimGCL	*PAAC*	LightGCN	SimGCL	*PAAC*
Gowalla	0.001233	0.001139	**0.000978**	0.010066	0.001202	**0.000585**
Yelp2018	0.000808	0.000718	**0.000523**	0.018836	0.001154	**0.000400**
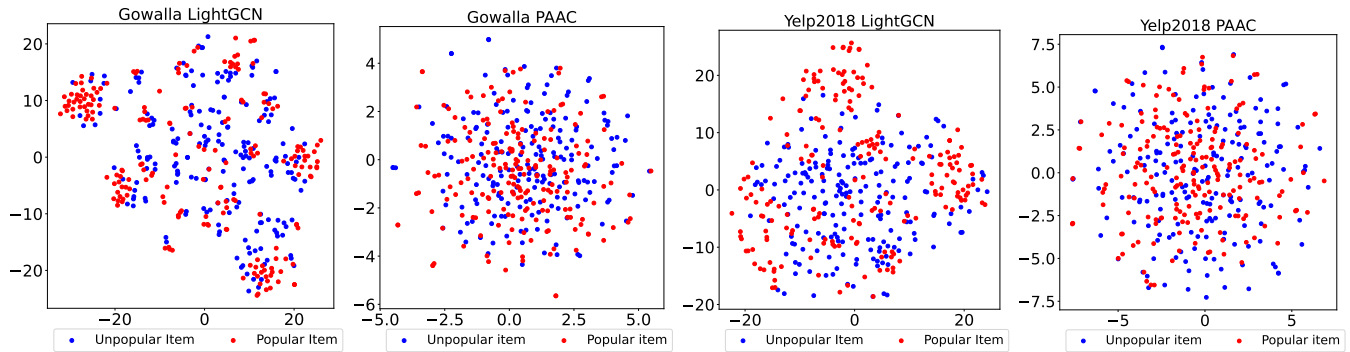
Figure 6: Visualization of Item Embeddings with t-SNE [29]: We present a t-SNE visualization of 400 randomly selected item embeddings from our dataset.

these groups, we employed two key metrics: Maximum Mean Discrepancy (MMD) and Cosine Similarity (CS). MMD measures the statistical distance between two distributions and is defined as:

$$\text{MMD}^2(X, Y) = \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(y_j) \right\|^2,$$

where X and Y represent samples from the two distributions, with m and n as their respective sample sizes, and ϕ denotes a feature map into a reproducing kernel Hilbert space. Conversely, Cosine Similarity (CS) evaluates the angular difference between vectors, reflecting the diversity in their representations:

$$\text{CS}(x_1, x_2) = \frac{x_1 \cdot x_2}{\|x_1\| \|x_2\|},$$

where x_1 and x_2 are the embedding vectors. Table. 7 shows the results for MMD and CS, demonstrating that *PAAC* significantly lowers the disparity between the distributions of popular and unpopular items—as evidenced by reduced values in both metrics—thereby effectively enhancing embedding separation.

Qualitative Analysis. The effectiveness of *PAAC* is further validated through a comprehensive visual assessment using t-SNE [29], a widely used technique for dimensionality reduction and visualization. This analysis involved visualizing the embeddings of a randomly selected subset of 400 items from the dataset. By plotting these embeddings, we are able to visually compare the performance of *PAAC* against a baseline model, LightGCN.

Figure 6 provides a clear illustration of this comparison. The figure demonstrates that PAAC achieves a significantly more uniform distribution of embeddings compared to LightGCN. In particular, both popular and unpopular items are more evenly dispersed throughout the embedding space under *PAAC*, rather than clustering in separate regions. This uniform distribution is indicative of reduced embedding separation, a crucial factor in mitigating popularity bias. The visual assessment, therefore, provides strong qualitative evidence that complements our statistical analyses, reinforcing the overall robustness and efficacy of *PAAC* in promoting fairer and more balanced representation of items in the embedding space.