Cross Pairwise Ranking for Unbiased Item Recommendation

Qi Wan
wqq17@mail.ustc.edu.cn
University of Science and Technology
of China
China

Xiangnan He*
xiangnanhe@gmail.com
University of Science and Technology
of China
China

Xiang Wang
xiangwang1223@gmail.com
University of Science and Technology
of China
China

Jiancan Wu
wjc1994@mail.ustc.edu.cn
University of Science and Technology
of China
China

Wei Guo
guowei67@huawei.com
Huawei Noah's Ark Lab
China

Ruiming Tang
tangruiming@huawei.com
Huawei Noah's Ark Lab
China

ABSTRACT

Most recommender systems optimize the model on observed interaction data, which is affected by the previous exposure mechanism and exhibits many biases like popularity bias. The loss functions, such as the mostly used pointwise Binary Cross-Entropy and pairwise Bayesian Personalized Ranking, are not designed to consider the biases in observed data. As a result, the model optimized on the loss would inherit the data biases, or even worse, amplify the biases. For example, a few popular items take up more and more exposure opportunities, severely hurting the recommendation quality on niche items — known as the notorious Mathew effect.

In this work, we develop a new learning paradigm named *Cross Pairwise Ranking* (CPR) that achieves unbiased recommendation without knowing the exposure mechanism. Distinct from inverse propensity scoring (IPS), we change the loss term of a sample — we innovatively sample multiple observed interactions once and form the loss as the combination of their predictions. We prove in theory that this way offsets the influence of user/item propensity on the learning, removing the influence of data biases caused by the exposure mechanism. Advantageous to IPS, our proposed CPR ensures unbiased learning for each training instance without the need of setting the propensity scores. Experimental results demonstrate the superiority of CPR over state-of-the-art debiasing solutions in both model generalization and training efficiency. The codes are available at https://github.com/Qcactus/CPR.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; **Recommender systems**.

---

*Xiangnan He is the corresponding author.

KEYWORDS

Unbiased Learning-to-Rank, Popularity Bias, Recommendation

1 INTRODUCTION

Recommender systems have become the core of many online platforms, including e-Commerce, streaming media, social networks etc. Existing models mostly follow a supervised learning paradigm, which treats historical interactions (*e.g.,* implicit feedback [15, 16, 28] like clicks and purchases) as the labeled data, and learns user-item relevance by fitting the labeled data. Two types of loss functions are intensively used to optimize the model parameters: (1) pointwise loss, which captures a user's preference on single item by minimizing the discrepancy between the target and predicted relevance score of a user-item pair, *e.g.,* Binary Cross-Entropy (BCE) [11, 16, 19] and Mean Squared Error (MSE) [17, 18]; and (2) pairwise loss, which models a user's preference on two items by encouraging the prediction of positive item to be higher than that of the negative item, *e.g.,* Bayesian Personalized Ranking (BPR) [28].

However, such standard loss functions are prone to the biases inherent in the observed interaction data. For example, the interaction data usually presents a long-tail distribution on item popularity [2], that is, a small number of popular items occupy most interactions. Recent studies show that item popularity confounds user true preference and observed interactions [37, 43] and should be properly handled for quality recommendation. Nonetheless, mainstream loss functions are designed to recover the historical data — assuming that observed interactions reflect user preference faithfully — while leaving the bias effect untouched. Hence, when building recommender systems with these losses, a few popular items take up more and more exposure opportunities, severely hurting the recommendation quality on niche items.

How to mitigate the bias and perform unbiased estimation of user preference has become one central theme in recommendation [7]. One prevalent solution is Inverse Propensity Scoring (IPS) [12, 21, 29, 30], which reweights each data sample by the inverse of its propensity score (i.e., the exposure probability). Despite that IPS

is unbiased in theory, it suffers from practical limitations: (1) it is challenging to accurately estimate the propensity score of each sample, since the exposure mechanism is seldom known [31]; and (2) the reweighted loss usually exhibits a high variance especially for implicit feedback [29, 30, 32], which implies that the losses of individual samples fluctuate heavily from their expected values.

In this work, we propose a sample-level unbiased learning paradigm, Cross Pairwise Ranking (CPR), without the need of quantifying the exposure mechanism. The key assumption is that the propensity score can be factorized into user propensity, item propensity, and user-item relevance. With this assumption, we can offset the impact of the user/item propensity by designing a novel loss function that combines purposefully the chosen user-item pairs. Briefly speaking, superior to IPS, the proposed CPR ensures unbiased learning for each individual sample and suits better for debiasing the learning from implicit feedback. Furthermore, for faster convergence in training, we borrow the idea of hard negative sampling [27, 42] and devise a dynamic sampling method for CPR.

In a nutshell, we summarize the contributions as follows:

- We analyze the commonly used pointwise and pairwise loss functions in recommender system from a new perspective, showing that they are biased in deriving the correct ranking of user preference.
- We propose a new learning method CPR and offer theoretical analyses of its unbiasedness in estimating user preference on individual samples.
- We conduct empirical studies on two benchmark datasets, validating the effectiveness of CPR in recommendation debiasing on multiple backbone models.

2 PRELIMINARIES

We first introduce the basic notations. Then we formulate a new perspective on the unbiased learning and analyze the biasedness of the pointwise and pairwise loss from this perspective.

2.1 Notations

Let $u \in \mathcal{U}$ be a user and $i \in \mathcal{I}$ be an item. We call a user-item pair a positive pair if we observe an interaction between them, and call it a negative pair otherwise.

Following the notations in previous literature [29, 30], we define three binary variables: the interaction variable $Y_{u,i}$, the relevance variable $R_{u,i}$ and the observation variable $O_{u,i}$. $Y_{u,i} = 1$ represents an interaction happens between user u and item i, and 0 otherwise. $R_{u,i} = 1$ means the user likes the item, and 0 otherwise. $O_{u,i} = 1$ means the user observes the item, and 0 otherwise. An interaction happens when the user likes and observes the item, i.e., $Y_{u,i} = R_{u,i} \cdot O_{u,i}$, leading to:

$$\begin{aligned} P(Y_{u,i} = 1) &= P(R_{u,i} = 1, O_{u,i} = 1) \\ &= P(R_{u,i} = 1)P(O_{u,i} = 1 \mid R_{u,i} = 1). \end{aligned} \quad (1)$$

$P(R_{u,i} = 1)$ indicates the relevance probability between user u and item i. $P(O_{u,i} = 1 \mid R_{u,i} = 1)$ is the exposure probability — the probability of item i being observed by user u conditioned on their relevance. Here we do not assume the independence between $R_{u,i}$ and $O_{u,i}$ as in previous studies [29, 30], since in the real world, users are more likely to be exposed to highly relevant items because of

the exposure mechanism of the previous recommender that also captures user-item relevance. Besides, due to the bias in the previous exposure mechanism, among all potentially relevant items, users are more likely to be exposed to the popular ones, i.e., $P(O_{u,i} = 1 \mid R_{u,i} = 1)$ of popular items is generally higher than that of niche items, which increases their interaction probability $P(Y_{u,i} = 1)$ and makes the recommendation lean towards them.

Let $s_{u,i}$ be $\ln P(R_{u,i} = 1)$. We call $s_{u,i}$ the true relevance score between user u and item i. More precisely, $s_{u,i} = \ln \left[a \cdot P(R_{u,i} = 1) \right]$, where a is a positive constant that changes the scale of $s_{u,i}$; we ignore this constant since it does not matter in the following analyses. We use $\hat{s}_{u,i}$ to denote the predicted relevance score, which is typically obtained by feeding the user and item embeddings into an interaction function like inner product or a neural network [39]. $\hat{s}_{u,i}$ is expected to estimate $s_{u,i}$, but the challenge is that we only have observation data on $Y_{u,i}$, whereas $R_{u,i}$ and $O_{u,i}$ are unobservable.

2.2 Definition of Unbiasedness

Previous studies on IPS [29, 30] define the unbiasedness of losses from the statistical view of expectation, making the loss expectation equal to the ideal loss by reweighting samples. However, it suffers from some practical limitations, such as the difficulty of setting propensity scores and the high variance of the reweighted loss. We leave the discussion to Section 4.1.

In this work, we propose a new view on the unbiasedness of loss, that is free from the limitations of IPS:

DEFINITION 2.1. *A loss function $\mathcal{L}$ is unbiased if it optimizes the ranking of predicted user-item relevance scores towards that of the true relevance scores:*

$$\hat{s}_{u,i} > \hat{s}_{u,j} \Leftrightarrow s_{u,i} > s_{u,j} \text{ when } \mathcal{L} \text{ converges.}$$

In the rest of this section, we will briefly introduce the traditional pointwise and pairwise losses, and analyze their biasedness from this new perspective.

2.3 Biasedness of Pointwise and Pairwise Loss

To learn user preference from historical interactions, most models follow the supervised learning paradigm, which encourages the relevance scores to recover the labels. Two standard loss functions are widely used to optimize the model parameters: (1) The pointwise loss like BCE [11, 16, 19] aims to capture the user preference on single items via minimizing the prediction discrepancy:

$$\mathcal{L}_{BCE} = - \sum_{(u,i) \in \mathcal{D}} Y_{u,i} \ln \sigma(\hat{s}_{u,i}) + (1 - Y_{u,i}) \ln(1 - \sigma(\hat{s}_{u,i})), \quad (2)$$

where $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$; $\mathcal{D}^+$ is the set of observed interactions, and $\mathcal{D}^-$ is the set of missing data, and $\sigma(\cdot)$ is the sigmoid function. (2) The pairwise loss, such as BPR [28], models the user preference on two items. It encourages the prediction of the positive item to be higher than that of the negative item:

$$\mathcal{L}_{BPR} = - \sum_{(u,i,j) \in \mathcal{D}_S} \ln \sigma(\hat{s}_{u,i} - \hat{s}_{u,j}), \quad (3)$$

where $\mathcal{D}_S = \{(u, i, j) \mid Y_{u,i} = 1, Y_{u,j} = 0\}$. Next, we analyze how they lead to the biased estimation of users' true preference.

2.3.1 Biasedness of Pointwise Loss.

PROPOSITION 2.1. *Pointwise loss is biased.*

PROOF. The pointwise loss in Equation (2) performs binary classification between the relevance scores of positive pairs and negative pairs, which can be represented by the following inequality:

$$\begin{cases} \hat{s}_{u,i} \geq C & \text{if } Y_{u,i} = 1, \\ \hat{s}_{u,i} < C & \text{if } Y_{u,i} = 0. \end{cases} \quad (4)$$

where C is the constant that denotes the classification threshold.

To maximize the likelihood of observing $Y_{u,i}$, we should rank $P(Y_{u,i} = 1)$ as follows:

$$\begin{cases} \ln P(Y_{u,i} = 1) \geq C & \text{if } Y_{u,i} = 1, \\ \ln P(Y_{u,i} = 1) < C & \text{if } Y_{u,i} = 0. \end{cases}$$

which can be rewritten as:

$$\begin{cases} s_{u,i} + \ln P(O_{u,i} = 1 \mid R_{u,i} = 1) \geq C & \text{if } Y_{u,i} = 1, \\ s_{u,i} + \ln P(O_{u,i} = 1 \mid R_{u,i} = 1) < C & \text{if } Y_{u,i} = 0. \end{cases} \quad (5)$$

This is the expected ranking of $s_{u,i}$. Compared with this desired ranking, Equation (4) actually uses $\hat{s}_{u,i}$ to model $s_{u,i} + \ln P(O_{u,i} = 1 \mid R_{u,i} = 1)$. The resultant $\hat{s}_{u,i}$ is generally higher for popular items, since their exposure probability tends to be higher. Therefore, the pointwise loss is biased, and models optimized with it will favor popular items. □

2.3.2 Biasedness of Pairwise Loss.

PROPOSITION 2.2. *Pairwise loss is biased.*

PROOF. The pairwise loss in Equation (3) encourages the scores of positive pairs to be higher than those of negative pairs for each user, which can be represented by the following inequality:

$$\hat{s}_{u,i} - \hat{s}_{u,j} > 0, \text{ if } Y_{u,i} = 1, Y_{u,j} = 0. \quad (6)$$

According to the expected ranking (5), we have:

$$s_{u,i} + \ln P(O_{u,i} = 1 \mid R_{u,i} = 1) \geq C, \text{ if } Y_{u,i} = 1;$$
$$s_{u,j} + \ln P(O_{u,j} = 1 \mid R_{u,j} = 1) < C, \text{ if } Y_{u,j} = 0.$$

By a subtraction between these two inequalities, we can write this ranking in a pairwise form:

$$s_{u,i} + \ln P(O_{u,i} = 1 \mid R_{u,i} = 1) - \left[s_{u,j} + \ln P(O_{u,j} = 1 \mid R_{u,j} = 1) \right] > 0, \text{ if } Y_{u,i} = 1, \ Y_{u,j} = 0. \quad (7)$$

Compared with this desired ranking, Equation (6) also uses $\hat{s}_{u,i}$ to model $s_{u,i} + \ln P(O_{u,i} = 1 \mid R_{u,i} = 1)$, which is generally higher for popular items, thus the pairwise loss is biased. □

3 METHOD

In this section, we propose a new loss named CPR, and theoretically prove its unbiasedness under a reasonable assumption. Then we extend this loss to a more general form for a stronger supervision on learning. Finally, we design a dynamic sampling algorithm for CPR to speed up the training and also improve its performance.

3.1 CPR Loss

Below we present the proposed CPR loss and discuss its rationale in the following subsections. We first construct a training sample by selecting two positive user-item pairs (u_1, i_1) and (u_2, i_2), so that $u_1 \neq u_2$, meanwhile (u_1, i_2) and (u_2, i_1) are negative pairs. On such training samples, we propose a novel loss as follows:

$$\mathcal{L}_{CPR} = - \sum_{(u_1, u_2, i_1, i_2) \in \mathcal{D}_2} \ln \sigma \left[\frac{1}{2} (\hat{s}_{u_1, i_1} + \hat{s}_{u_2, i_2} - \hat{s}_{u_1, i_2} - \hat{s}_{u_2, i_1}) \right] \quad (8)$$

where $\mathcal{D}_2 = \{(u_1, u_2, i_1, i_2) \mid Y_{u_1, i_1} = 1, Y_{u_2, i_2} = 1, Y_{u_1, i_2} = 0, Y_{u_2, i_1} = 0\}$ denotes the training data.

3.2 Unbiasedness of CPR Loss

CPR loss encourages the sum score of the two positive pairs to be higher than that of the two negative pairs, which are composed by crossing the two pairs, *i.e.,*

$$\hat{s}_{u_1, i_1} + \hat{s}_{u_2, i_2} - \hat{s}_{u_1, i_2} - \hat{s}_{u_2, i_1} > 0,$$
$$\text{if } Y_{u_1, i_1} = 1, Y_{u_2, i_2} = 1, Y_{u_1, i_2} = 0, Y_{u_2, i_1} = 0. \quad (9)$$

To demonstrate its unbiasedness, we first make a mild assumption: the exposure probability can be factorized into user propensity, item propensity, and user-item relevance, which is formulated as:

$$P(O_{u,i} = 1 \mid R_{u,i} = 1) = p_u \cdot p_i \cdot P(R_{u,i} = 1)^\alpha, \quad (10)$$

where p_u and p_i are user-specific and item-specific propensity, respectively, typically higher for active users and popular items; $P(R_{u,i} = 1)^\alpha$ reflects that higher relevance contributes more to the exposure probability; α is a positive constant for smoothing purpose. With this assumption, the expected ranking in Equation (5) can be rewritten as:

$$\begin{cases} s_{u,i} \geq \frac{1}{1+\alpha}(C - \ln p_u - \ln p_i) & \text{if } Y_{u,i} = 1, \\ s_{u,i} < \frac{1}{1+\alpha}(C - \ln p_u - \ln p_i) & \text{if } Y_{u,i} = 0. \end{cases} \quad (11)$$

We can revisit the pointwise ranking in Equation (4), which is biased against this correct ranking owing to the impact of p_u and p_i. Similarly, by rewriting the correct pairwise ranking in Equation (7), we find that the pairwise ranking in Equation (6) is biased against this correct ranking due to the impact of p_i:

$$s_{u,i} - s_{u,j} > \frac{1}{1+\alpha}(-\ln p_i + \ln p_j), \text{ if } Y_{u,i} = 1, Y_{u,j} = 0. \quad (12)$$

PROPOSITION 3.1. *CPR loss is unbiased under the assumption of Equation* (10).

PROOF. Given the expected ranking under the assumption, we inspect one training sample involving the positive pairs (u_1, i_1) and (u_2, i_2) and the negative pairs (u_1, i_2) and (u_2, i_1), and have:

$$s_{u_1, i_1} \geq \frac{1}{1+\alpha}(C - \ln p_{u_1} - \ln p_{i_1}), \ s_{u_2, i_2} \geq \frac{1}{1+\alpha}(C - \ln p_{u_2} - \ln p_{i_2}),$$

$$s_{u_1, i_2} < \frac{1}{1+\alpha}(C - \ln p_{u_1} - \ln p_{i_2}), \ s_{u_2, i_1} < \frac{1}{1+\alpha}(C - \ln p_{u_2} - \ln p_{i_1}).$$

By a subtraction between the sum of the first two inequalities and the sum of the last two, we get:

$$s_{u_1, i_1} + s_{u_2, i_2} - s_{u_1, i_2} - s_{u_2, i_1} > 0,$$
$$\text{if } Y_{u_1, i_1} = 1, \ Y_{u_2, i_2} = 1, \ Y_{u_1, i_2} = 0, \ Y_{u_2, i_1} = 0. \quad (13)$$
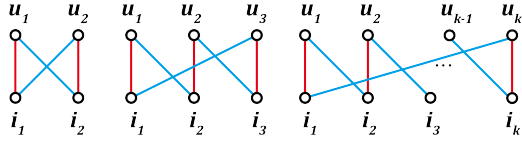
Figure 1: Composition of CPR samples. Red lines denote positive pairs. Blue lines denote negative pairs. Graphs from left to right denote the composition of samples in $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_k$, respectively.

p_u and p_i are canceled here, since they have the same total impact on the positive pairs as on the negative pairs. Clearly, CPR ranking (Equation (9)) coincides with this form of expected ranking— $s_{u,i}$ is successfully modeled by $\hat{s}_{u,i}$. Thus, CPR loss is unbiased. $\quad\square$

3.3 Extending to More Interactions

The CPR loss discussed above only uses two observed interactions as a training sample. We now extend it to $k(k \geq 2)$ interactions, while preserving its unbiasedness:

$$\mathcal{L}_{CPR} = -\sum_k \sum_{(u_1,\ldots,u_k,i_1,\ldots,i_k)\in\mathcal{D}_k}$$

$$\ln\sigma\left[\frac{1}{k}\cdot(\hat{s}_{u_1,i_1}+\cdots+\hat{s}_{u_k,i_k}-\hat{s}_{u_1,i_2}-\cdots-\hat{s}_{u_k,i_1})\right] \quad (14)$$

where $\mathcal{D}_k = \{(u_1,\ldots,u_k,i_1,\ldots,i_k) \mid Y_{u_1,i_1} = 1, Y_{u_2,i_2} = 1,\ldots, Y_{u_k,i_k} = 1, Y_{u_1,i_2} = 0, Y_{u_2,i_3} = 0,\ldots,Y_{u_k,i_1} = 0\}$, $k \geq 2$, and $\frac{1}{k}$ scales the loss of samples with different numbers of interactions. We omit the detailed proof of its unbiasedness, which is similar to the previous section. Figure 1 illustrates the composition of samples with different numbers of interactions.

These different types of samples set different restrictions to the ranking, so it is a natural idea to set k as many values in the loss. But in experiments, CPR usually achieves the best performance when $k = 2, 3$, and the performance cannot be improved when k takes more values greater than 3. One possible reason is that a larger k makes the training inflexible — it forces more pairs to be trained together, even when some of them are about to overfit. In the experiments described later, samples are only drawn from $\mathcal{D}_2$ and $\mathcal{D}_3$.

The complete loss function used for CPR is:

$$\mathcal{L} = \mathcal{L}_{CPR} + \lambda(\|\Theta\|^2), \quad (15)$$

where Θ denotes model parameters, λ is the L_2 regularization coefficient to control overfitting.

3.4 Dynamic Sampling

Obviously, how to construct the sample set $\mathcal{D}_k$ plays a critical role in CPR. One straightforward solution is to perform a random sampling: first draw k non-overlap positive user-item pairs, then accept them as one sample if their cross-combinations are all negative pairs, otherwise discard them and redraw. In this way, all samples are selected with the same probability. However, during the training, some hard samples [27] might take more iterations to converge, while some easy samples have less contribution to the model optimization and should be avoided.

Inspired by DNS [42], which is a dynamic negative sampling strategy for BPR [28], we devise a dynamic sampling strategy for CPR, with the aim of assigning hard samples with higher sampling probability and training them more frequently. Algorithm 1 details how to dynamically select a batch of samples. We use b, $\beta(\beta \geq 1)$ and $\gamma(\gamma > 1)$ to denote the batch size, the dynamic sampling rate and the choosing rate, respectively. First we randomly select $b\beta\gamma$ samples, each of which contains k observed interactions (line 2). The choosing rate γ is to increase the number of initial samples to ensure that after discarding inappropriate samples in the following step, we can still collect the desired number of samples. Next, we discard the samples whose cross-combinations are not k negative pairs, and obtain $b\beta$ useful samples (line 3-6). We measure the difficulty of a sample by the value of $\frac{1}{k}\cdot(\hat{s}_{u_1,i_1}+\hat{s}_{u_2,i_2}+\cdots+\hat{s}_{u_k,i_k}-\hat{s}_{u_1,i_2}-\hat{s}_{u_2,i_3}-\cdots-\hat{s}_{u_k,i_1})$ — the smaller the value is, the harder it is for this sample to reach its target ranking. Therefore, we select samples with the least values of this expression as a batch of samples (line 7-10).

Algorithm 1: Dynamic Sampling

Input: Batch size b, dynamic sampling rate $\beta \geq 1$, choosing rate $\gamma > 1$, parameter k

Output: A batch of samples $\mathcal{D}_k^{batch}$ for training

1   $\mathcal{S} \longleftarrow \emptyset$

2   Randomly select $b\beta\gamma$ samples, each of which contains k positive user-item pairs

3   **for** *each sample* $\{(u_1,i_1),(u_2,i_2),\ldots,(u_k,i_k)\}$ **do**

4   **if** $\{(u_1,i_2),(u_2,i_3),\ldots,(u_k,i_1)\}$ *is a set of negative pairs* **then**

5       $\mathcal{S} \longleftarrow \mathcal{S} \cup (u_1,\ldots,u_k,i_1,\ldots,i_k)$

6   **if** $|\mathcal{S}| = b\beta$ **then** break

7   Calculate

$O_j = \frac{1}{k}\cdot(\hat{s}_{u_1,i_1}+\hat{s}_{u_2,i_2}+\cdots+\hat{s}_{u_k,i_k}-\hat{s}_{u_1,i_2}-\hat{s}_{u_2,i_3}-\cdots-\hat{s}_{u_k,i_1})$

for $S_j = (u_1,\ldots,u_k,i_1,\ldots,i_k) \in \mathcal{S}$

8   Sort $S_1,\ldots,S_{b\beta}$ by the ascending order of $O_1,\ldots,O_{b\beta}$

9   $\mathcal{D}_k^{batch} \longleftarrow$ the first b samples in the sorted list

10   **return** $\mathcal{D}_k^{batch}$

---

4 DISCUSSION

We compare CPR with related work to demonstrate the superiority of its novel design. First we compare with a prevalent debiasing method, IPS, and show how CPR avoids the difficulty of setting propensity scores and the high-variance problem. Then we discuss the difference between CPR and Setwise Ranking [8, 26, 33].

4.1 Comparison with IPS

IPS and CPR are both debiasing approaches that focus on the loss function, but with different views on the unbiasedness of loss and hence provide different solutions. Here we briefly explain the methodology of IPS and how our CPR avoids or mitigates the inherent problems of IPS.

Let $\mathcal{L}_{ideal}$ be the ideal loss function where the exposure mechanism is properly handled. According to IPS theory, the loss function $\mathcal{L}$ is unbiased if its expectation across the probabilistic distribution

of exposure is equal to the ideal loss:

$$\mathbb{E}_{O_{i,j}}(\mathcal{L}(\hat{s}_{u,i})) = \mathcal{L}_{ideal}(\hat{s}_{u,i})$$

With this definition, the authors prove that traditional losses are biased for having an expectation not equal to the ideal loss. To deal with this problem, they reweight the label $Y_{u,i}$ used in the traditional loss functions by dividing it by a *propensity score*, which is an estimate of $P(O_{i,j} = 1)$. However, two problems emerge from this solution:

- The propensity score is typically calculated by a power-law function of the item degree (the number of its occurrences in the training set), which is an oversimplification of the exposure mechanism.
- Although the reweighted loss can have an ideal expectation (if we ignore the inaccuracy of propensity scores for now), its variance can be large (Theorem 4.4 in [30], Theorem 3.4 in [29], and [32]), in other words, for each sample, the reweighted loss deviates from the ideal loss. Variance reduction techniques are usually applied, such as clipping the weight or the loss of each sample to a certain range; but in this way, the loss expectation is no longer unbiased.

Our CPR alleviates these two problems:

- CPR takes a more general assumption on the exposure mechanism, *i.e.,* Equation (10), without explicitly setting any propensity score.
- CPR changes the composition of loss instead of reweighting the original loss, and thus achieves sample-level unbiasedness that is well guaranteed under the mild assumption of $P(O_{u,i} = 1 \mid R_{u,i} = 1)$.

CPR loss is *not* perfect — it ensures the unbiasedness at the cost of pursuing a relaxed optimization goal: encouraging the sum score of k ($k \geq 2$) positive pairs to be higher than that of k negative pairs, which may not ensure the expected one-to-one comparison in Definition 2.1. But this relaxation on optimization can be empirically justified by the good performance of CPR.

4.2 Comparison with Setwise Ranking

Setwise Ranking [8, 26, 33] is another line of work that also make adaptations on BPR. But CPR is quite different from it in terms of both the motivation and the method.

- *Motivation.* Setwise Ranking aims to improve the performance of BPR by ranking item sets instead of instances. It is not designed for debiasing purpose.
- *Method.* For Setwise Ranking, given a user u, two sets are constructed: $\mathcal{S}_u^+$ is a set of some positive items and $\mathcal{S}_u^-$ is a set of some negative items. Then the relevance between $\mathcal{S}_u^+$ and u is encouraged to be closer than that of $\mathcal{S}_u^-$ and u. In other words, they still use triplets $\{(u, \mathcal{S}_u^+, \mathcal{S}_u^-)\}$ for pairwise ranking. But CPR uses a more complex form of sample $\{(u_1, \ldots, u_k, i_1, \ldots, i_k) \mid Y_{u_1,i_1} = 1, Y_{u_2,i_2} = 1, \ldots, Y_{u_k,i_k} = 1, Y_{u_1,i_2} = 0, Y_{u_2,i_3} = 0, \ldots, Y_{u_k,i_1} = 0\}$ for cross pairwise ranking, which endows CPR with the debiasing ability that Setwise Ranking methods do not have.

Table 1: Data statistics.

Dataset	#Users	#Items	#Interactions	Density
MovieLens	61,770	6,958	1,533,956	0.0036
Netflix	46,420	12,898	2,475,020	0.0041
iFashion	264,251	53,033	1,505,141	0.0001

5 EXPERIMENTS

We first describe experimental settings, and then evaluate our proposed method as compared to other competitive methods. We also examine the debiasing effect and generalization ability of CPR, and study the effects of different compositions of CPR samples.

5.1 Experimental Settings

5.1.1 Datasets. We conduct empirical studies on three real-world datasets: MovieLens-10M [14], Netflix Prize [3] and Alibaba iFashion [9]. Ratings in MovieLens and Netflix are binarized by setting five-star ratings to 1 and the rest to 0. For iFashion, the user clicks on fashion outfits are regarded as interactions. For all datasets, We adopt 3-core setting to keep more unpopular items instead of the commonly used 10-core setting that discards users and items with less than 10 interactions [28], since the debiasing performance largely depends the performance on unpopular items. The statistics of the processed datasets are summarized in Table 1.

5.1.2 Evaluation Protocol. To evaluate the proposed debiasing approach, we follow the offline evaluation protocol in previous work [5, 22, 44] to create simulated unbiased data. More specifically, we sample the records from the full dataset with equal probability in terms of items to create the validation and testing sets, leaving the rest as training data. Moreover, we cap the sampling probability with an upper limit, otherwise, the probability can be large for some unpopular items, making these items over sampled into the validation/testing set, while leaving few in the training set. More formally, let $p_s(u, i)$ be the probability of sampling the user-item pair (u, i), we set $p_s(u, i) \propto \min(\frac{1}{d_i}, a)$, where d_i is the number of occurrences of i in the full dataset, and the upper limit a is set to $\frac{1}{60}$ for MovieLens and Netflix, and set to $\frac{1}{12}$ for the more sparse dataset iFashion. The sampled data is randomly split into a validation set and a test set, and the rest data is treated as the training set. We obtain a 70/10/20 split for the training/validation/testing sets.

We measure the performance by three evaluation metrics: Recall@K, NDCG@K and ARP@K, where K is set to 20 by default. Recall@K and NDCG@K are widely used to measure the quality of the recommendations; since here we adopt an unbiased evaluation protocol, these metrics are calculated on the simulated unbiased sets where popular items are downsampled, emphasizing the importance of unpopular items, so higher values of these metrics reflect both higher quality and better unbiasedness of the recommendation. Average Recommendation Popularity at K (ARP@K) introduced by [1, 41] is an additional measure for (un)biasedness. It calculates the average popularity of the top-K recommended items for each user, where the item popularity is represented by its degree. Lower values of ARP@K reflect better unbiasedness of the recommendation.

5.1.3 Baselines. We refer to the approaches that do not explicitly handle popularity bias as traditional approaches, and those that handle the bias as debiasing approaches. We compare the proposed

CPR with two traditional approaches (the first two below) and four debiasing approaches (the rest four):

- **BPR** [28]: BPR is a commonly used training method, which optimizes the personalized ranking by minimizing the pairwise ranking loss.
- **Mult-VAE** [23]: This is a recommendation model based on the variational autoencoder (VAE), which assumes the data is generated from a multinomial distribution and uses variational inference for parameter estimation.
- **CausE** [5]: This is a domain adaptation method, which assigns two embeddings to each item — one is learned on a large amount of biased data and the other is learned on a small amount of unbiased data. It further regularizes them to be similar.
- **Rel-MF** [30]: It is an IPS method that modifies the traditional pointwise loss.
- **UBPR** [29]: It is an IPS method that modifies the traditional pairwise loss. It uses a non-negative loss function to reduce the variance of the estimator.
- **DICE** [44]: It is the state-of-the-art debiasing method. For each user and item, it learns two disentangled embeddings for two causes, user interest and conformity, by sampling cause-specific data for training.

Except for Mult-VAE, which must use the VAE architecture and loss, other approaches, as well as our CPR can be applied across different backbones. For a fair comparison, we adopt the simple and widely used model, Matrix Factorization (MF) [20], as the backbone of these approaches to conduct all the experiments expect for those in Section 5.4.1, where the aim is to compare the approaches under different backbones including LightGCN [15], NeuMF [16], and NGCF [35].

5.1.4 hyperparameter Settings. We implement CPR and baselines in TensorFlow. The embedding size is 128 for all methods except for DICE. To keep the number of embedding parameters consistent among all the methods, we set the embedding size to 64 for DICE since it learns two embeddings for each user and item. All the models are trained with Adam optimizer via early stopping. For CPR, samples are drawn from both $\mathcal{D}_2$ and $\mathcal{D}_3$ to compute the loss (set $k = 2, 3$ in Equation (14)) and the sampling ratio of $\mathcal{D}_2$ and $\mathcal{D}_3$ is tuned on the validation set; we will empirically justify this sample composition and provide hyperparameter study on the sampling ratio in Section 5.5. The dynamic sampling rate in CPR is also tuned on the validation set; the choosing rate is fixed to 2.

5.1.5 Significance Test. To detect significant differences between CPR and the best baseline on each dataset, we repeat their experiments 5 times by changing the random seed. The two-tailed pairwise t-test is performed using their results.

5.2 Overall Comparison

Experimental results demonstrate the superiority of CPR in terms of both effectiveness and efficiency.

5.2.1 Effectiveness Comparison. The effectiveness comparison is presented in Table 2. Our methods are CPR-rand and CPR. CPR-rand adopts random sampling while CPR performs dynamic sampling. CPR-rand beats all the baselines on MovieLens and Netflix. On
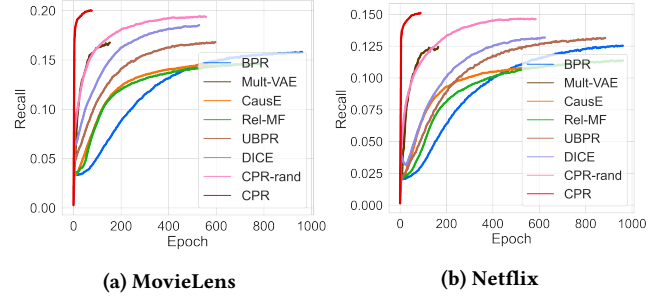

(a) MovieLens　　　**(b) Netflix**

Figure 2: Comparison of Recall curves.
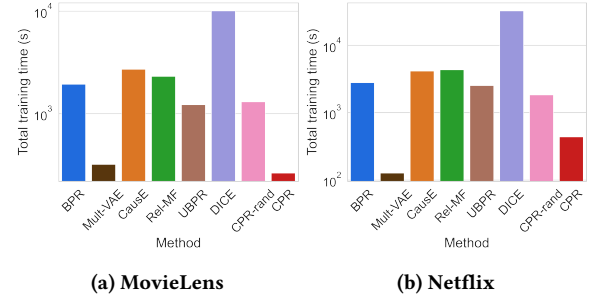

(a) MovieLens　　　**(b) Netflix**

Figure 3: Comparison of the total training time.

iFashion, CPR-rand achieves similar Recall with the best baseline, but with a lower ARP which indicates that its recommendation is less biased. CPR further improves the performance of CPR-rand on all datasets, significantly outperforming all baselines on Recall and NDCG, with p-values less than 0.05, which indicates that the improvement achieved by CPR over the best baseline is statistically significant. Note that the Recall and NDCG are measured on the simulated unbiased test sets where popular items are downsampled, it is normal to find their values relatively low compared to those in the papers where the test sets are randomly sampled [23], because models generally perform worse on unpopular items.

The higher Recall and NDCG of CPR compared to other approaches on unbiased test set indicates its superior prediction accuracy, especially on unpopular items. The lower ARP of CPR further indicates that CPR tends to recommend more unpopular items, in other words, it can better mitigate the popularity bias.

5.2.2 Efficiency Comparison. Figure 2 shows the Recall curves of all approaches on MovieLens and Netflix, where it can be observed that CPR converges to the best performance with the least number of epochs, contributing by the dynamic sampling. Although dynamic sampling processes more samples to select the relatively difficult ones, it can be implemented efficiently with negligible cost. Figure 3 compares the total training time of all approaches; the y-axis is log-scaled. The strongest baseline DICE requires the longest training time, while our CPR is relatively fast among all approaches. More specifically, CPR takes only 2.6% and 1.3% of the training time of DICE on MovieLens and Netflix, respectively.

5.3 Distribution of Recommendation

In order to examine the debiasing effect of CPR more intuitively, we compare the percentage of recommended items grouped by their degrees for each approach, along with their percentage in the training and test sets.

Table 2: Overall Performance. The best result in each column is in bold; the best baseline result in each column is underlined. Significance test is conducted on Recall and NDCG between CPR and the best baseline. ⋆ : p-value < 0.05. %Improv.: percentage of improvement on Recall and NDCG of CPR over the best baseline. Higher Recall and NDCG means higher prediction accuracy; lower ARP means less popularity bias.

Method	Backbone	MovieLens			Netflix			iFashion		
		Recall	NDCG	ARP	Recall	NDCG	ARP	Recall	NDCG	ARP
BPR	MF	0.1579	0.0939	4084	0.1255	0.0952	3341	0.0278	0.0122	436
Mult-VAE	-	0.1676	0.1004	4468	0.1242	0.0921	3554	<u>0.0309</u>	<u>0.0142</u>	415
Debiasing approaches:										
CausE		0.1440	0.0805	3814	0.1080	0.0730	3206	0.0214	0.0094	408
Rel-MF	MF	0.1463	0.0829	4107	0.1138	0.0765	3254	0.0250	0.0113	421
UBPR		0.1682	0.1000	2691	0.1315	0.1007	2289	0.0281	0.0126	404
DICE		<u>0.1835</u>	<u>0.1101</u>	<u>1173</u>	<u>0.1317</u>	<u>0.1007</u>	<u>1244</u>	0.0275	0.0124	**250**
CPR-rand	MF	0.1938	0.1192	**1055**	0.1462	0.1143	1270	0.0307	0.0137	398
CPR		**0.2003**⋆	**0.1223**⋆	1138	**0.1511**⋆	**0.1190**⋆	**1204**	**0.0332**⋆	**0.0151**⋆	359
%Improv.		9.16%	11.08%	-	14.73%	18.17%	-	7.44%	6.34%	-

(a) MovieLens (b) Netflix
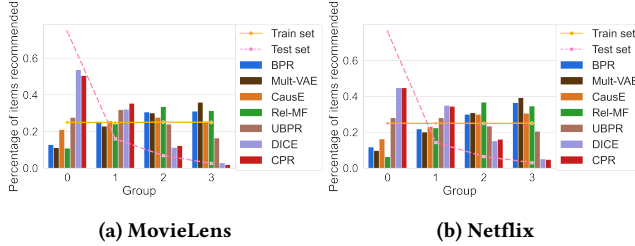
Figure 4: Comparison of percentage of recommended items of each group.

We categorize all items into four groups in the following way: sorting all items by their degrees in the ascending order, and grouping them such that the sum of degrees in each group are approximately equal. Figure 4 shows the percentage of degrees contained in each group of the training set and the test set, and the percentage of the recommended items of each group by each approach. The curve of the training set is flat due to our grouping method. While for the testing set, the percentage of recommended items decreases as the group ID increases since popular items are downsampled, as a result, low-degree items take a larger proportion. For an ideal unbiased recommendation, its distribution is expected to match that of the test set.

BPR and Mult-VAE both amplify the bias in the training set, recommending more popular items from Group 2 and 3. Debiasing approaches like UBPR, DICE and CPR alleviate the bias by recommending more unpopular items from Group 0 and 1. CPR and DICE achieve the best debiasing result, making the distribution closer to that of the test set.

5.4 Generalization Ability

5.4.1 Different Backbones. We apply our method CPR, as well as BPR and the best baseline DICE on MovieLens and Netflix, to different backbones to evaluate their generalization ability. We adopt the state-of-the-art graph model LightGCN [15], a neural model NeuMF

Table 3: Performance *w.r.t.* different backbones.

Method	Backbone	MovieLens		Netflix	
		Recall	NDCG	Recall	NDCG
BPR		0.1661	0.0995	0.1310	0.0985
DICE	LightGCN	0.1864	0.1121	0.1355	0.1043
CPR		**0.1963**	**0.1210**	**0.1502**	**0.1180**
BPR		0.1519	0.0890	0.1254	0.0932
DICE	NeuMF	0.1568	0.0914	0.1126	0.0841
CPR		**0.1895**	**0.1155**	**0.1434**	**0.1109**
BPR		0.1615	0.0951	0.1199	0.0899
DICE	NGCF	0.1774	0.1071	0.1312	0.0999
CPR		**0.1952**	**0.1182**	**0.1498**	**0.1173**

[16], and another graph model NGCF [35] as different backbones. Results in Table 3 verify that CPR consistently outperforms BPR and DICE on these backbones. For example, on MovieLens, CPR performs better than DICE by 7.9%, 26.4% and 10.4% *w.r.t.* NDCG@20 on LightGCN, NeuMF and NGCF, respectively.

5.4.2 Different Degrees of Bias. We also evaluate the generalization ability of CPR using training data with different degrees of bias. We sample 70% of the training set with probability $p_s(u, i) \propto d_i^\theta$, where d_i is the item degree in the training set, and θ is set to 0.5 and -0.5, respectively. When $\theta = 0.5$, the bias in the sampled training set is amplified compared with the original training set; when $\theta = -0.5$, the bias is reduced. Table 4 shows that CPR consistently outperforms the best baselines, DICE, with around 10% and 14% improvement *w.r.t.* NDCG@20 on MovieLens and Netflix, respectively with both sampled training sets ($\theta = 0.5$ and $\theta = -0.5$).

5.5 Composition of Samples

We compare the performances of CPR with different sample compositions, *i.e.*, different k values in Equation (14). We first fix k to single values: $k = 1$, $k = 2$, or $k = 3$ and tune other hyperparameters to obtain the best performances under each value of k. The dashed lines in Figure 5 show that the value of Recall decreases as

Table 4: Performance *w.r.t.* different degrees of bias.

θ	Method	MovieLens		Netflix	
		Recall	NDCG	Recall	NDCG
	BPR	0.1217	0.0679	0.0939	0.0654
0.5	DICE	0.1461	0.0839	0.1048	0.0741
	CPR	**0.1584**	**0.0927**	**0.1189**	**0.0858**
	BPR	0.1413	0.0809	0.1086	0.0777
-0.5	DICE	0.1597	0.0956	0.1133	0.0858
	CPR	**0.1737**	**0.1040**	**0.1274**	**0.0965**

(a) MovieLens

(b) Netflix
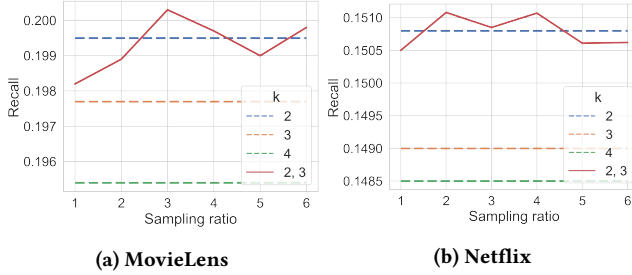
Figure 5: Comparison of sample compositions.

k increases, but not much — the performance under $k = 4$ is still better than all baselines in Table 2. As we discussed in Section 3.3, the performance drop is probably due to the inflexibility of training that comes with the larger k. Next, we set $k = 2, 3$. We choose the sampling ratio of $\mathcal{D}_2$ and $\mathcal{D}_3$ from $[1, 2, 3, 4, 5, 6]$, and tune other hyperparameters under each value of the sampling ratio. Results are drawn with solid lines in Figure 5. The performance under $k = 2, 3$ can be better than that under $k = 2$ with the appropriate sampling ratio which is typically around 3, indicating that different types of samples can be used in combination to achieve better results compared to using a single type of samples.

6 RELATED WORK

Generally, there are three research lines for unbiased recommendation on implicit feedback. Here we review each of them.

6.0.1 Domain Adaptation. Some studies perform domain adaptation by utilizing a small amount of unbiased data as the target domain to guide the training on biased source data. For example, CausE [5] corrects the biased embedding by minimizing its distance from another embedding learned on the unbiased data. KDCRec [24] uses knowledge distillation to transfer the knowledge obtained from biased data to the modeling of unbiased data. However, since the unbiased data is often rare, their performance is limited in real scenarios. To avoid the usage of unbiased data, ESAM [10] treats displayed and non-displayed items as source and target domains, respectively, and minimizes the distance between their covariance matrices. However, domain adaptation methods often suffer from loss of domain-specific information — some important item features learned from biased data may be lost when aligning with the features from unbiased data.

6.0.2 Counterfactual Learning. Counterfactual learning methods utilize causal graphs to handle the effect of item popularity on recommendation. DICE [44] uses two separate embeddings for each user and item to model the causal effects of user-item relevance and exposure mechanism, respectively, by training with cause-specific data. MACR [37] and CR [34] removes the direct effect of item properties on predicted scores by causal inference. DIB [25] and PDA [43] both remove the confounding popularity bias during training, but PDA [43] further inject the future popularity to the scores during inference. Unlike our proposed method that focuses on constructing an unbiased loss function, these approaches deal with the popularity bias from a different point of view — they analyze the causal effect between the bias and the observed data and then apply causal operations accordingly.

6.0.3 Inverse Propensity Score (IPS). IPS is a prevalent debiasing method for recommender systems [22, 31], and recently explored in the implicit recommendation setting. Rel-MF [30] is the first IPS method designed for implicit feedback, which reweights the pointwise loss by the estimated item exposure probability to obtain an expectation-unbiased loss function. Later, UBPR [29] is proposed to extend the pointwise model in Rel-MF to a pairwise version. UEBPR [12] and DU [21] introduce neighborhood-based explainability and unclicked data reweighting to the plain IPS methods, respectively. AutoDebias [6] combines IPS with data imputation and adopts a meta-learning algorithm to learn the optimal debiasing configurations on a small uniform data. These IPS methods mainly have two drawbacks: the difficulty in estimating propensity scores and the high variance of the reweighted loss.

7 CONCLUSION AND FUTURE WORK

In this work, we revisit the commonly used pointwise and pairwise loss functions from a new perspective, and point out that they are biased in approaching the correct ranking of user preference. Under the assumption that the exposure probability can be factorized as user propensity, item propensity, and user-item relevance, we ascribe the biasedness in traditional loss functions to user/item propensity. We then propose a new loss, CPR, where the impact of the user/item propensity is offset by the combination of carefully selected user-item pairs, so as to approach the unbiased ranking of user preference. Experimental results show that CPR achieves better debiasing performance on multiple backbone models.

This initial work sheds a light on mitigating the popularity bias without the knowledge of the exposure mechanism, with the help of a mild assumption. One limitation of CPR is that more interactions in one sample would bring inflexibility to training, which prevents us from using more types of samples together to further improve the recommendation accuracy. We hope to solve this problem in future work. Another limitation is that the proposed assumption of the exposure probability may not reflect the real exposure mechanism accurately. We hope to propose a more general assumption in future work. Moreover, we would like to extend CPR to other scenarios, such as exploring its ability to alleviate the group fairness bias [4], combining it with list ranking [13] or graph learning [36, 38, 40].

ACKNOWLEDGMENTS

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-ranking. In *FLAIRS*.

[2] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided exposure bias in recommendation. In *IRS@KDD*.

[3] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *KDDCup*. 35.

[4] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *KDD*. 2212–2220.

[5] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *RecSys*. 104–112.

[6] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. In *SIGIR*. 21–30.

[7] 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. *arXiv preprint arXiv:2010.03240* (2020).

[8] Lei Chen, Le Wu, Kun Zhang, Richang Hong, and Meng Wang. 2021. Set2setRank: Collaborative Set to Set Ranking for Implicit Feedback Based Recommendation. In *SIGIR*. 585–594.

[9] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In *KDD*. 2662–2670.

[10] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. Esam: Discriminative domain adaptation with non-displayed items to improve long-tail performance. In *SIGIR*. 579–588.

[11] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *DLRS*. 7–10.

[12] Khalil Damak, Sami Khenissi, and Olfa Nasraoui. 2021. Debiased Explainable Pairwise Ranking from Implicit Feedback. In *RecSys*. 321–331.

[13] Saurabh Gupta, Bharathan Balaji, and Runfei Luo. 2020. CPR: Collaborative Pairwise Ranking for Online List Recommendations.. In *ORSUM@ RecSys*.

[14] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *TIIS* 5, 4 (2015), 1–19.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[17] Xiangnan He, Jinhui Tang, Xiaoyu Du, Richang Hong, Tongwei Ren, and Tat-Seng Chua. 2019. Fast matrix factorization with nonuniform weights on missing data. *TNNLS* 31, 8 (2019), 2791–2804.

[18] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*. 263–272.

[19] Christopher C Johnson. 2014. Logistic matrix factorization for implicit feedback data. In *NeurIPS Workshop on Distributed Machine Learning and Matrix Computations*.

[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[21] Jae-woong Lee, Seongmin Park, and Jongwuk Lee. 2021. Dual Unbiased Recommender Learning for Implicit Feedback. In *SIGIR*. 1647–1651.

[22] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *UAI Workshop on Causation: Foundation to Application*.

[23] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *WWW*. 689–698.

[24] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *SIGIR*. 831–840.

[25] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. Mitigating Confounding Bias in Recommendation via Information Bottleneck. In *RecSys*. 351–360.

[26] Weike Pan and Li Chen. 2013. Cofiset: Collaborative filtering via learning pairwise preferences over item-sets. In *SDM*. 180–188.

[27] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*. 273–282.

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.

[29] Yuta Saito. 2020. Unbiased Pairwise Learning from Biased Implicit Feedback. In *ICTIR*. 5–12.

[30] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *WSDM*. 501–509.

[31] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*. PMLR, 1670–1679.

[32] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *NeurIPS*. 3231–3239.

[33] Chao Wang, Hengshu Zhu, Chen Zhu, Chuan Qin, and Hui Xiong. 2020. SetRank: A setwise Bayesian approach for collaborative ranking from implicit feedback. In *AAAI*, Vol. 34. 6127–6136.

[34] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*. 1288–1297.

[35] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.

[36] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat seng Chua. 2021. Towards Multi-Grained Explainability for Graph Neural Networks. In *NeurIPS*.

[37] Tianxin Wei, Fuli Feng, Jiawei Chen, Chufeng Shi, Ziwei Wu, Jinfen Yi, and Xiangnan He. 2021. Model-agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. In *KDD*.

[38] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR*. 726–735.

[39] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2021. A Survey on Neural Recommendation: From Collaborative Filtering to Content and Context Enriched Recommendation. *arXiv preprint arXiv:2104.13030* (2021).

[40] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. In *ICLR*.

[41] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the Long Tail Recommendation. In *VLDB*.

[42] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*. 785–788.

[43] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR*.

[44] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Depeng Jin, and Yong Li. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *WWW*.