

Fairly Adaptive Negative Sampling for Recommendations

Xiao Chen
The Hong Kong Polytechnic
University and Center for Artificial
Intelligence and Robotics, HKISI CAS
shaw.n.chen@connect.polyu.hk

Haochen Liu
Michigan State University
liuhaoc1@msu.edu

Wenqi Fan*
The Hong Kong Polytechnic
University
wenqifan03@gmail.com

Zitao Liu
Guangdong Institute of Smart
Education, Jinan University
liuzitao@jnu.edu.cn

Jingfan Chen
The Hong Kong Polytechnic
University and Center for Artificial
Intelligence and Robotics, HKISI CAS
jingfan.chen@connect.polyu.hk

Zhaoxiang Zhang*
The Hong Kong Polytechnic
University; Center for Artificial
Intelligence and Robotics, HKISI CAS;
and Institute of Automation, Chinese
Academy of Sciences
zhaoxiang.zhang@ia.ac.cn

Qing Li
The Hong Kong Polytechnic
University
csqli@comp.polyu.edu.hk

ABSTRACT

Pairwise learning strategies are prevalent for optimizing recommendation models on implicit feedback data, which usually learns user preference by discriminating between positive (i.e., clicked by a user) and negative items (i.e., obtained by negative sampling). However, the size of different item groups (specified by item attribute) is usually unevenly distributed. We empirically find that the commonly used uniform negative sampling strategy for pairwise algorithms (e.g., BPR) can inherit such data bias and oversample the majority item group as negative instances, severely countering group fairness on the item side. In this paper, we propose a **Fairly** adaptive **Negative** sampling approach (**FairNeg**), which improves item group fairness via adaptively adjusting the group-level negative sampling distribution in the training process. In particular, it first perceives the model's unfairness status at each step and then adjusts the group-wise sampling distribution with an adaptive momentum update strategy for better facilitating fairness optimization. Moreover, a negative sampling distribution Mixup mechanism is proposed, which gracefully incorporates existing importance-aware sampling techniques intended for mining informative negative samples, thus allowing for achieving multiple optimization purposes. Extensive experiments on four public datasets show our proposed method's superiority in group fairness enhancement and fairness-utility tradeoff.

*Corresponding authors



This work is licensed under a Creative Commons Attribution International 4.0 License.

WWW '23, April 30–May 04, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9416-1/23/04.
<https://doi.org/10.1145/3543507.3583355>

CCS CONCEPTS

• Information Systems → Recommender systems.

KEYWORDS

Recommender Systems, BPR, Fairness, Negative Sampling.

ACM Reference Format:

Xiao Chen, Wenqi Fan, Jingfan Chen, Haochen Liu, Zitao Liu, Zhaoxiang Zhang, and Qing Li. 2023. Fairly Adaptive Negative Sampling for Recommendations. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*, April 30–May 04, 2023, Austin, TX, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3543507.3583355>

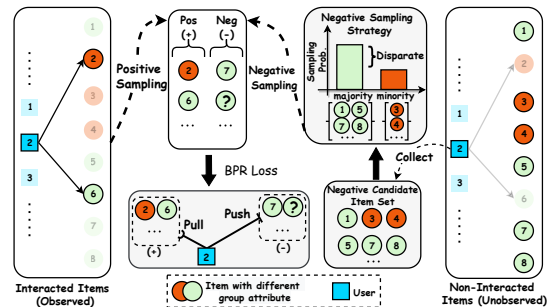


Figure 1: Illustration of bias in Uniform Negative Item Sampling. There are two groups of items (determined by sensitive attributes, such as job payment or movie genres). Unobserved items in the majority group (in green color) have a higher probability of being sampled as negative items than that of the minority item group (in red color).

1 INTRODUCTION

Personalized recommender systems have been widely deployed to alleviate information overload issues [14, 15] by providing users with relevant items on various online services [8, 13, 15, 50], such

as online shopping, job matching, online education, etc. As the most representative techniques in recommender systems, Collaborative Filtering (CF) methods [3, 10] aim to learn recommendation models by encoding user-item implicit feedback (e.g., user clicks and purchases) with a pairwise learning paradigm, e.g., Bayesian Personalized Ranking (BPR) [7, 38]. In general, such a mainstream learning paradigm first conducts randomly negative item sampling to construct training data [12, 22], where it usually pairs a positive item (from observed user interactions) with a negative item (from unobserved interactions), and then encourages the predicted score of the positive item to be higher than that of the negative item.

However, such a mainstream learning paradigm is vulnerable to data bias [2, 28] on the item side, considering that the size of different item groups (determined by a specific item attribute) is highly imbalanced in the real world. For example, *low-paying* jobs (majority group) occupy a larger portion than *high-paying* jobs (minority group) in the job matching system, and the number of *comedy* movies overpasses the *Horror* movies for movie recommendations, both of which can be viewed as the inherent data bias on the item side [18]. Moreover, to optimize recommendation models via the BPR loss, the prevalent negative sampling technique, i.e., Uniform Negative Sampling (UNS), is more likely to treat unobserved items in the majority group as negative.

For instance, as illustrated in Figure 1, items in the majority group (in green color) gain higher sampling probability as negative instances than that of the minority group (in red color) when learning for user u_2 's preference. As a result, in the training process, the majority item group will be pushed to obtain low (biased) prediction scores via the BPR loss, resulting in disparate recommendation performance (e.g., Recall) over different item groups. In other words, such a uniform negative sampling strategy naturally inherits group-wise data bias on the item side, which misleads recommendation models into learning biased users' preferences and dampens the recommendation quality for items in the majority group.

To address such undesired bias and unfairness caused by the negative sampling procedure, a straightforward solution is to adjust the negative sampling distribution by equalizing each item group's chance of being sampled as negative and positive instances, such that items in the majority group will not be oversampled as negative instances for the BPR optimization. Here we call such a bias mitigation method as **FairStatic**. In order to verify the aforementioned unfair treatment over different item groups in UNS and its corresponding solution (i.e., FairStatic), we conduct preliminary studies to examine how these two sampling strategies (i.e., UNS v.s. FairStatic) perform on two recommendation methods: a classic Matrix Factorization model (MF) [38] and an advanced Graph Neural Network-based model (LightGCN) [11, 22]. For simplicity, we only consider the two-group scenario and work on the ML1M-2 dataset, whose movie items can be categorized into two genres (i.e., Sci-Fi and Horror). The detailed data description is shown in Experiments Section 4, and the experimental results are shown in Figure 2.

It can be observed that MF/LightGCN+UNS methods exhibit a significant disparity in recommendation performance between the two item groups under Recall@20 metric, while MF/LightGCN+FairStatic could reduce the group-wise disparity at the cost of sacrificing overall recall performance. From these observations, we conclude that adjusting the negative item sampling distribution can

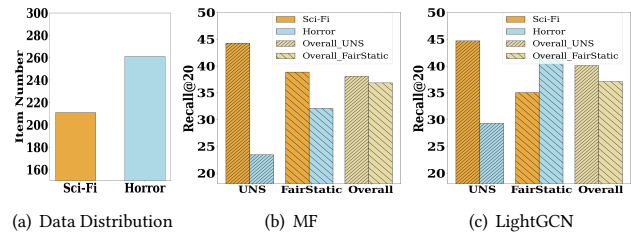


Figure 2: (a) Distribution of two item groups on the ML1M-2 dataset. In (b) and (c), the Recall@20 performance of Sci-Fi and Horror movie groups are reported on two recommendation models with two different negative sampling strategies.

alleviate group unfairness on the item side. However, such naive adjustment based on the static (i.e., fixed in the whole training process) distribution greedily pursues the fairness objective, easily leading to a sub-optimal recommendation performance. Therefore, it is desirable to design an adaptive negative sampling method for achieving a better tradeoff between item group fairness and overall recommendation accuracy (utility).

In this paper, we seek to address item-side group performance unfairness in recommendations from a novel perspective. Existing negative sampling strategies used in the pairwise learning paradigm usually neglect the item's group attribute [4, 6], which makes them vulnerable to item-side data bias and exhibit disparate group performance. Correspondingly, we propose a novel fairly adaptive negative sampling method (**FairNeg**), which can dynamically adjust the group-level negative sampling distribution based on the perceived performance disparity during the model training process. Moreover, FairNeg can be gracefully merged with importance-aware sampling techniques and yield a better utility-fairness tradeoff. Our major contributions are summarized as follows:

- To the best of our knowledge, we are the first to investigate the bias and unfairness issues in recommendations from the perspective of negative sampling. We empirically show a principled connection between item-oriented group performance fairness and the group-level negative sampling distribution;
- We propose a novel **Fairly** adaptive **Negative** sampling framework (**FairNeg**) to mitigate item group unfairness in recommendations, where the group-level negative sampling distribution is updated adaptively via an outer optimizer, and the standard recommendation model is kept as the inner optimizer. These two optimizers altogether turn the training paradigm into a bi-level optimization framework, which enables easier-to-reconfigure implementation for fair recommendations;
- Extensive experiments on different real-world datasets demonstrate the effectiveness of the proposed methods in significantly mitigating performance disparity among different item groups while maintaining the user-side overall recommendation quality.

2 PRELIMINARIES

In this section, we introduce some preliminary knowledge concerning the problem under study. We first present the notions related to the top-k item recommendation task [51] and the formulation of BPR loss. Then we introduce the definition and evaluation metric

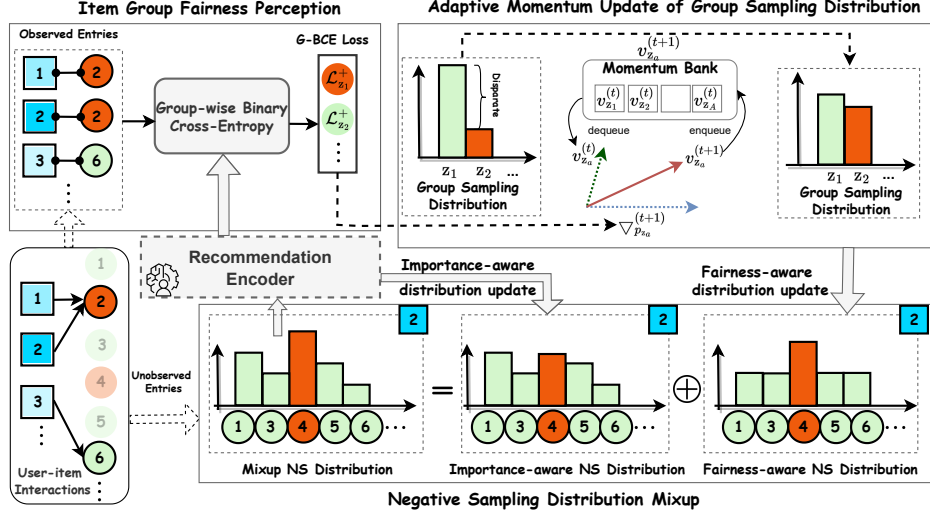


Figure 3: Illustration of the proposed FairNeg framework. There are three components in this Fairly adaptive Negative sampling (FairNeg) framework: (i) Item Group Fairness Perception, (ii) Adaptive Momentum Update of Group Sampling Distribution, and (iii) Negative Sampling (NS) Probability Distribution Mixup.

of item-oriented group performance fairness, which deserves our efforts in this paper.

2.1 Personalized Ranking on Implicit feedback

Let $\mathcal{U} = \{u_1, \dots, u_m\}$, $\mathcal{V} = \{v_1, \dots, v_n\}$ be the sets of users and items respectively, and $Y \in \mathbb{Y}^{m \times n}$ to represent the user-item historical (implicit) interaction matrix, where $y_{ui} = 1$ (in Y) represents an observed interaction between user u and item i , and 0 otherwise. In addition, we use \mathcal{V}_u^+ to denote the interacted (positive) item set of user u , and $\mathcal{V}_u^- = \mathcal{V} - \mathcal{V}_u^+$ to represent the un-interacted (negative) item set. In this paper, we focus on the top- K item recommendation task based on the implicit interactions Y , which can be generally optimized by the pairwise BPR loss \mathcal{L}_{BPR} for learning model's parameters Θ . The main idea is to encourage the predicted scores on observed (positive) interactions to be higher than those of unobserved (negative) items, which can be formalized as follows:

$$\min_{\Theta} \mathcal{L}_{\text{BPR}} = - \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{V}_u^+, j \in \mathcal{V}_u^-} \ln \sigma(\hat{y}_{u,i} - \hat{y}_{u,j}), \quad (1)$$

where the negative item j is sampled from the unobserved item set \mathcal{V}_u^- of user u , $\hat{y}_{u,i}$ and $\hat{y}_{u,j}$ denote the predicted ranking scores generated by the recommendation model $f(\cdot)$ with parameters Θ , and σ is the sigmoid function. Here we ignore the regularization term for simplicity.

2.2 Problem Definition

We assume a set of item groups with size A , $Z = \{z_1, z_2, \dots, z_A\}$, specified by the item attribute (e.g., job payment, ethnicity, or movie genre). These groups are non-overlapping since each item belongs to one group. In this paper, we focus on enhancing **Item-oriented Group Performance Fairness** of recommendation models.

In particular, to evaluate whether recommendation models treat different item groups equally, in the test stage, we can measure each group's recommendation performance using metrics such as $\text{Recall}@k$, which is an essential metric that reflects the coverage of truly relevant items in the top- k recommendation results for

one item group. Then, the disparity of group-wise $\text{Recall}@k$ performance (abbreviated as $\text{Recall-Disp}@k$) can be calculated following the definition of previous work [52]:

$$\text{Recall-Disp}@k = \frac{\text{std}(\text{Recall}@k|z = z_1, \dots, \text{Recall}@k|z = z_A)}{\text{mean}(\text{Recall}@k|z = z_1, \dots, \text{Recall}@k|z = z_A)}, \quad (2)$$

where $\text{std}(\cdot)$ is the standard deviation function and $\text{mean}(\cdot)$ calculates the mean average value. Such relative standard deviation can be computed to reflect the group-wise performance disparity. The item-oriented fairness (indicated by the lower value of $\text{Recall-Disp}@k$) is supposed to be fulfilled in general item recommendation systems so that all items, regardless of their group attributes, have the same chance of being exposed to users who truly like them, and no user needs are ostensibly ignored.

3 THE PROPOSED METHOD

In this section, we will first give an overview of the proposed method FairNeg, then describe each of its components in detail and finally illustrate the designed optimization algorithm for item group fairness.

3.1 An Overview of the Proposed Framework

Since the empirical risk minimization (i.e., BPR loss) does not consider the fairness criterion, the recommendation model usually exhibits severe unfairness issues. To address such limitations without explicitly changing the model architecture or training data, we design a fairly adaptive negative sampling framework FairNeg, as shown in Figure 3, which adjusts each item group's negative sampling probability for fairness considerations and incorporates importance-aware technique for enhancing negative samples' informativeness. The proposed FairNeg method consists of three main components: (i) Item Group Fairness Perception, (ii) Adaptive Momentum Update of Group Sampling Distribution, and (iii) Negative Sampling Distribution Mixup. The first component, Item Group Fairness Perception, is designed to perceive unfairness (i.e.,

recall disparity) among item groups during the training process. Next, with the perceived group performance disparity, the second component (i.e., Adaptive Momentum Update of Group Sampling Distribution) adjusts *each group's negative sampling probability* accordingly, intending to equalize all groups' recall performance. At last, the third component (i.e., Negative Sampling Distribution Mixup) further incorporates an *importance-aware* sampling probability with a mixup mechanism. In this way, the group fairness optimization and the informativeness of negative samples can be jointly considered.

3.2 Item Group Fairness Perception

To achieve fair training from the perspective of adaptive negative sampling, we first need to perceive the group-wise performance disparity [19, 32, 39], which provides guidance for adjusting the group-level negative sampling distribution in the next step. However, the fairness evaluation metric (i.e., Recall-Disp@k) mentioned above cannot be used here *directly* in the training phase since it is non-differentiable. We hereby propose a Group-wise Binary Cross-Entropy (G-BCE) loss as a proxy to measure each item group's recall performance. It only considers observed interactions (viewed as positive) in the training set and calculates the group-wise average classification loss as an approximation of group recall [19]. Next, we will introduce the details of G-BCE loss, followed by the analysis revealing that mitigating the disparity of *multiple* groups' G-BCE losses at the training stage actually optimizes item-oriented group fairness.

More specifically, G-BCE loss measures the average difference between the predicted preference scores $\hat{y}_{u,v}$ and the truly observations $y_{u,v}$ in a specific item group z_a :

$$\begin{aligned} \mathcal{L}_{z_a}^+ &= - \frac{1}{|Y_{z_a}^+|} \sum_{(u,v) \in Y_{z_a}^+} \underbrace{(y_{u,v} \cdot \log \sigma(\hat{y}_{u,v}))}_{\text{Positive Part}} \\ &\quad + \underbrace{(1 - y_{u,v}) \cdot \log(1 - \sigma(\hat{y}_{u,v}))}_{\text{Negative Part}} \\ &= - \frac{1}{|Y_{z_a}^+|} \sum_{(u,v) \in Y_{z_a}^+} \log \sigma(\hat{y}_{u,v}), \end{aligned}$$

where $Y_{z_a}^+$ denotes the set of all true observations for item group z_a , $\hat{y}_{u,v}$ denotes the predicted preference score of user u to item v , and σ is the sigmoid function that normalizes the preference score to (0, 1). Note that the negative part in Equation 3 can be eliminated, since we only consider observed entries (referring to $y_{u,v}$ equals to 1) and $\log(\sigma(\hat{y}_{u,v}))$ is equivalent to $\log(p(\sigma(\hat{y}_{u,v}) = 1))$. As a result, $\mathcal{L}_{z_a}^+ = -\frac{1}{|Y_{z_a}^+|} \sum_{(u,v) \in Y_{z_a}^+} \log \sigma(\hat{y}_{u,v})$ can be viewed as an unbiased estimation of $-\log p(\hat{y} = 1 \mid y = 1, z = z_a)$, which further approximates $-\log(\text{Recall} \mid z = z_a)$.

Taking the two groups scenario as an example, if $\mathcal{L}_{z_1}^+ > \mathcal{L}_{z_2}^+$ in the training phase, it indicates that the item group z_1 (disadvantaged group) obtains lower recall performance. In other words, the recommendation model assigns a lower recommendation probability to items in group z_1 , given that users truly like these items. Next, we focus on minimizing the disparity of G-BCE loss (Group-wise Recall) during training, which is equivalent to optimizing toward item-oriented group fairness (as previously defined).

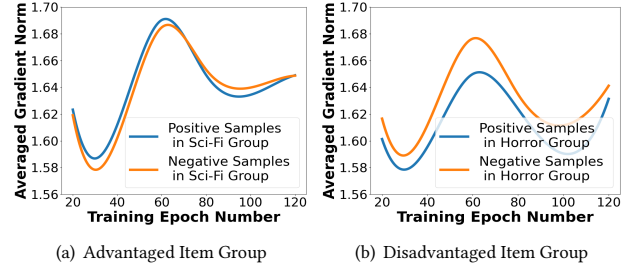


Figure 4: The overall gradient analysis on positive and negative samples in two item groups on the ML1M-2 dataset (as described in Table 1) and the backbone is a Matrix Factorization model. We collect the averaged L2 norm of the gradient of weights [43] in the item embedding layer at each epoch until model convergence.

3.3 Adaptive Momentum Update of Group Sampling Distribution

In this work, we propose to achieve group fairness from a novel perspective – negative sampling, which requires no change of model architecture or the training data. We attribute the dampening of the majority item group's recall performance (as shown in Figure 2) to the overwhelmed negative gradients, which is caused by the oversampling of being negatives of this group.

Concretely, when different groups' item size is fairly imbalanced, the majority item group is likely to be assigned with a relatively higher negative sampling probability. Note that in BPR loss (as shown in Equation 1), the paired negative item will receive discouraging gradients. Thus, the accumulated negative gradients of the majority group will make them obtain low predicted relevance scores, regardless of users' true preference on this group.

To verify this, we conduct an empirical analysis to show the *overwhelming negative gradients* on the majority item group during the optimization process, as illustrated in Figure 4. It is clear that the disadvantaged group (i.e., Horror group with worse recall performance) receives larger gradients from negative samples due to the high negative sampling probability of this group.

Based on the above analysis, we are inspired to design the adaptive negative sampling strategy in two consequent steps, where the (dis)advantaged group is first identified based on the perceived G-BCE losses (as illustrated in Section 3.2), and then the negative sampling probability of this group will be updated so that the disadvantaged group is less overwhelmed by negative gradients.

Specifically, we assign each item group a negative sampling probability and update it via (1) *Group-wise Gradient Calculation*, which calculates the current step's gradient of each group's sampling probability, and (2) *Adaptive Momentum Update*, which utilizes the auxiliary knowledge from historical gradients to alleviate optimization oscillations.

Group-wise Gradient Calculation. Considering items with a multi-valued sensitive attribute (i.e., $A \geq 2$), we define the group-wise negative sampling probability distribution as:

$$p := (p_{z_1}, p_{z_2}, \dots, p_{z_A}), \text{ where } \sum_{p_{z_a} \in \mathcal{P}} p_{z_a} = 1. \quad (3)$$

To narrow the performance gap among different groups, as illustrated in the previous analysis, we consistently adjust each group's

sampling probability \mathbf{p} according to its deviation from the averaged G-BCE loss at each step:

$$\nabla_{\mathbf{p}_{z_a}}^{(t)} := \mathcal{L}_{z_a}^{+(t)} - \frac{1}{|A|} \sum_{z \in Z} \mathcal{L}_z^{+(t)}, \quad (4)$$

where $\nabla_{\mathbf{p}_{z_a}}^{(t)}$ denotes the gradient of p_{z_a} at step t , $\mathcal{L}_{z_a}^{+(t)}$ denotes the G-BCE loss of group z_a at step t , and $\frac{1}{|A|} \sum_{z \in Z} \mathcal{L}_z^{+(t)}$ denotes the step t 's averaged G-BCE loss. Intuitively, a disadvantaged item group with a relatively large G-BCE loss (low recall performance) will receive a negative-valued update vector, which down-weights its negative sampling probability during the next step's training. Such an adaptive down-weighting scheme prevents the disadvantaged group from being overwhelmed by negative gradients and is therefore expected to boost the group's recall performance.

Adaptive Momentum Update. Furthermore, to alleviate the adverse influence of gradient noise [41] and instability of gradient direction, we design an adaptive momentum update strategy to produce the sampling probability of each item group at step $t + 1$, which can be summarized into two steps:

$$\begin{aligned} v_{z_a}^{(t+1)} &= \gamma v_{z_a}^{(t)} + \alpha \cdot \nabla_{\mathbf{p}_{z_a}}^{(t+1)}, \\ \mathbf{p}_{z_a}^{(t+1)} &= \mathbf{p}_{z_a}^{(t)} - v_{z_a}^{(t+1)}, \end{aligned} \quad (5)$$

where current step's momentum update vector $v_{z_a}^{(t+1)}$ is a combination of last step's momentum vector $v_{z_a}^{(t)}$ and current step's gradient $\nabla_{\mathbf{p}_{z_a}}^{(t+1)}$, $\gamma \in [0, 1]$ is the momentum coefficient, α is the learning rate; group-wise sampling probability $p_{z_a}^{(t+1)}$ at step $t + 1$ is then updated with the momentum update vector $v_{z_a}^{(t+1)}$. Here we maintain the *momentum bank* [21] as a queue of momentum update vectors: all groups' corresponding update vectors at the current step are enqueued, and the previous ones are dequeued. Intuitively, with the adaptive momentum update strategy, if the current epoch's gradient is in the same direction as the previous epoch, the updated value will be larger and helps to achieve faster convergence. By contrast, if the gradient direction conflicts, the updated value will be reduced, which facilitates smoothing out the variations and stable learning [5].

3.4 Negative Sampling Distribution Mixup

Though updating group sampling distribution adaptively ensures that item group fairness can be jointly optimized during training, the informativeness difference of items within the same group is neglected due to the equal-share of sampling probability. To be more specific, when drawing a negative sample from the unobserved item set \mathcal{V}_u^- for user u only following the updated group sampling distribution, for a certain candidate item j from the group z_a , its fairness-aware sampling probability can be computed as:

$$p_j^{fair} = \frac{p_{z_a}}{\sum_{i \in |\mathcal{V}_u^-|} G_{z_a}(i)}, \quad (6)$$

where p_{z_a} is the group-wise sampling probability, and $G_{z_a}(i)$ is a indicator function to identify whether item i in the negative candidate item set (unobserved) belongs to group z_a . The indicator function returns 1 when item j 's group attribute is z_a , otherwise 0. It can be seen that, in the candidate item set of user u , items from the same sensitive group share equal sampling probability. Nonetheless, such a sampling strategy tends to sample negative

items that are easier to distinguish from positive samples and inhibit improvement of user and item feature discriminativeness.

To address this limitation, we propose a Negative Sampling Distribution Mixup mechanism, which incorporate an importance-aware negative sampling strategy. Such a strategy is built upon the dynamic negative sampling method for BPR [49]. The importance of a candidate item refers to its relevance to the user. Formally, the importance-aware negative sampling probability of candidate item j for the user u can be calculated as follows:

$$p_j^{imp} = \frac{\exp(s_{u,j}/\tau)}{\sum_{i \in |\mathcal{V}_u^-|} \exp(s_{u,i}/\tau)}, \quad (7)$$

where $s_{u,j}$ is the user-item relevance score calculated by the recommendation model, and τ is a temperature parameter in the softmax function. The goal is to pick the high-scored item with a higher probability for user u , which facilitates learning more discriminative feature representations.

Now that the importance-aware negative sampling distribution (\mathbf{p}^{imp}) implicitly improves recommendation accuracy by strengthening feature representations, and the fairness-aware negative sampling distribution (\mathbf{p}^{fair}) aids item-oriented group fairness. To incorporate them simultaneously, we introduce a mixup hyper-parameter β , which controls the strength of fairness optimization. The final sampling probability p_j of candidate item j for user u can be formulated as:

$$p_j = \beta \cdot p_j^{fair} + (1 - \beta) p_j^{imp}. \quad (8)$$

With the new mixup negative sampling distribution computed for each user, the recommendation model training considers both item group fairness and user's preference learning.

3.5 Optimization for Item Group Fairness

We formulate the optimization of FairNeg as a bi-level problem, where the optimization of the group-wise negative sampling distribution is nested within the recommendation model parameters optimization. This can be solved by alternatively updating the fairness-aware negative group sampling distribution and recommendation model parameters [42]. More specifically, the negative sampling distribution is learned with the outer optimizer, while a personalized ranking algorithm (i.e., BPR) is kept as the inner optimizer for updating user and item representations.

Mathematically, such bi-level optimization can be formulated as follows:

$$\begin{aligned} \mathbf{p}^* &= \arg \min_{\mathbf{p}} \mathcal{L}_{\text{Recall-Disp}}(\Theta, \mathbf{p}) := \sum_{z_a \in Z} \left| \mathcal{L}_{z_a}^+ - \frac{1}{|A|} \sum_{z \in Z} \mathcal{L}_z^+ \right|, \\ \Theta^* &= \arg \min_{\Theta} \mathcal{L}_{\text{utility}}(\Theta, \mathbf{p}) := - \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{V}_u^+, j \in \mathcal{V}_u^-} \mathcal{L}_{\text{BPR}}(u, i, j; \Theta, \mathbf{p}), \end{aligned} \quad (9)$$

where Θ is the recommendation model parameters, and \mathbf{p} is the group-wise negative sampling distribution. $\mathcal{L}_{\text{Recall-Disp}}$ represents the disparity of G-BCE loss, which is the sum of each item group's deviation from the macro-average performance.

The optimization algorithm is presented in Algorithm 1. First, we initialize group-wise negative sampling distribution \mathbf{p} as the way of FairStatic, where each group's sampling probability equals its proportion in the overall user-item positive interactions; recommendation model parameters are initialized with Xavier [17] (line 1). Next, with \mathbf{p} fixed, we conduct pairwise learning and iteratively

update the recommendation model parameters Θ using BPR loss for one epoch. The negative items in the training pairs are sampled based on a mixup distribution, considering both item importance and group fairness (lines 2-7). Then, we fix the model parameters Θ and adjust each group’s sampling probability to reduce their recall disparity (lines 8-9). The iteration will stop early when the recommendation performance does not improve for consecutive p epochs (line 10).

Algorithm 1: Bi-level optimization of FairNeg

Input: Train data $(\mathcal{U}, \mathcal{V}, Y)$, Epochs E
Output: Optimized recommendation model parameter θ^*

- 1 Initialize model parameters θ and group-wise negative sampling distribution $\mathbf{p} := (p_{z_1}, p_{z_2}, \dots, p_{z_A})$;
- 2 **for each epoch do**
- 3 **for each** $(u, i) \in Y^+$ **do**
- 4 Obtain the candidate item set \mathcal{V}_u^+ for negative sampling;
- 5 Calculate each candidate item’s mixup sampling probability p_j according to Equation 6, 7, 8;
- 6 Sample a negative item j according to the mixup sampling distribution;
- 7 Conduct pairwise learning and update model parameters θ with (u, i, j) based on equation 1;
- 8 Calculate G-BCE loss for each item group z_a ;
- 9 Update each group’s sampling probability p_{z_a} based on Equation 4, 5;
- 10 Early stop when the model performance on validation set doesn’t improve for consecutive p epochs;
- 11 **return;**

4 EXPERIMENT

In this section, we conduct extensive experiments on four real-world datasets to demonstrate the effectiveness of our proposed FairNeg.

4.1 Experiment Settings

4.1.1 Datasets. The experiments are performed on publicly available datasets – MovieLens 1M (**ML1M**) [20] and **Amazon** product reviews [31]. The ML1M dataset consists of movie ratings in different genres, and we treat all ratings as positive feedback. The Amazon dataset includes the historical product reviews collected from the Amazon e-commerce platform, and we view all user purchase behaviors as positive feedback. Considering that some comparison methods can only work for the dataset with binary item attributes, we extract two subsets from each dataset, containing either a binary-valued or a multi-valued item attribute. Specifically, we extract ratings with four genres (‘Sci-Fi’, ‘Adventure’, ‘Children’, and ‘Horror’) to constitute our **ML1M-4** dataset, and keep the product reviews with four types (‘Grocery’, ‘Office’, ‘Pet’, and ‘Tool’) to constitute our **Amazon-4** dataset. Moreover, we create **ML1M-2** (‘Sci-Fi’ vs. ‘Horror’) and **Amazon-2** (‘Grocery’ vs. ‘Tool’) datasets by keeping the most popular (denoted by the average feedback number of this group) and least popular two groups. The statistics

Table 1: Characteristics of the datasets.

	Users	Items	Ratings	Density
ML1M-2	4,680	472	194,610	8.81%
ML1M-4	5,120	685	242,419	6.91%
Amazon-2	3,845	2,487	84,656	0.89%
Amazon-4	3,954	3,061	129,433	1.07%

of the four datasets are detailed in Table 1, and the detailed group-wise statistics (i.e., feedback and item number) are illustrated in the Appendix Section A.1. All datasets are randomly split into 60%, 20%, and 20% to constitute the training, validation, and test sets.

4.1.2 Evaluation Metrics. We evaluate the recommendation utility and the item group fairness in the experiments. For group fairness evaluation, we stratify the items by groups based on the item attribute, then compute **Recall@k** for each group based on the Top- k recommendation results. Afterward, we report three group fairness metrics as in the literature [24]: (i) **Recall-Disp**: the Recall disparity among all groups, and the formula is described in Equation 2. (ii) **Recall-Min**: the minimum Recall among all groups. (iii) **Recall-Avg**: the macro-average Recall over all groups. Here, the lower Recall-Disp value represents better group fairness, and the higher values for the latter two metrics are better. For the recommendation utility evaluation on the user side, we employ standard Top- k ranking metrics, including **Precision (P@k)**, **Recall(R@k)** score, and **Normalized Discounted accumulated Gain (N@k)**, measuring on top- k recommendation results [9, 46].

4.1.3 Parameter Settings. Due to the limited space, more implementation details are presented in the Appendix Section A.2.

4.1.4 Baselines. We compare FairNeg with four representative negative sampling methods and three state-of-the-art item-side fairness-aware recommendation methods. The method descriptions are as follows:

- **UNS** [38]: a widely used negative sampling approach using a static uniform distribution in the sampling process.
- **NNCF** [37]: a static negative sampling approach using a item-popularity-based distribution.
- **DNS** [49]: a dynamic negative sampling method that assigns higher probability to samples with higher ranking order.
- **FairStatic**: a simplified variant of FairNeg based on a static sampling distribution, as described in the introduction.
- **Reg** [23]: a typical debiasing method for datasets with a binary-valued attribute, which penalizes the squared difference between the average scores of two groups for all positive user-item pairs.
- **DPR** [52]: an advanced item group fairness method, which uses score-based adversarial-learning to decouple the predicted user-item preference score with the item’s attribute.
- **FairGAN** [26]: a GAN-based algorithm that fairly allocates exposure to each item with high user utility preservation.

4.2 Performance Comparison of Recommender Systems

The comparison experiments are conducted on two backbone models (**MF** and **LightGCN**) separately. Each experiment is conducted five times with different random seeds, and we report the average result. Based on top-20 recommendation results, the item group fairness and their recommendation performance on four datasets are respectively shown in Table 2, 3. The Reg method can only

Table 2: Fairness and recommendation utility of different methods on two backbone models over datasets with a binary-valued item attribute (evaluated on top-20 recommendation results). RI denotes the relative improvement of FairNeg over UNS. We highlight the best results in bold and the second best results with underline.

		ML1M-2						Amazon-2					
Backbone	Method	Recall-Disp.@20	Recall-Min.@20	Recall-Avg.@20	N@20	P@20	R@20	Recall-Disp.@20	Recall-Min.@20	Recall-Avg.@20	N@20	P@20	R@20
MF	UNS	0.3179	0.2346	0.3386	0.3487	<u>0.1611</u>	0.4622	0.6882	0.0386	0.1238	0.0780	0.0196	0.1471
	NNCF	0.1689	0.2550	0.3069	0.3051	0.1386	0.4157	<u>0.5157</u>	<u>0.0594</u>	0.1227	0.0727	0.0187	0.1159
	DNS	0.3065	0.2485	<u>0.3584</u>	0.3765	0.1702	0.4974	0.5937	0.0547	0.1347	0.0851	0.0209	0.1591
	FairStatic	0.0958	0.3208	0.3548	0.3409	0.1558	0.4541	0.5936	0.0501	0.1232	0.0769	0.0191	0.1426
	Reg	0.1908	0.2808	0.3471	0.3156	0.1580	0.4518	0.5648	0.0534	0.1227	0.0751	0.0189	0.1409
	DPR	0.0959	0.3147	0.3481	0.3134	0.1528	0.4295	0.5687	0.0480	0.0963	0.0610	0.0162	0.1283
	FairGAN-1	0.3607	0.2056	0.3217	0.3267	0.1495	0.4287	0.6503	0.0467	0.1285	0.0672	0.0167	0.1219
	FairNeg	0.0287	0.3659	0.3764	0.3540	0.1609	0.4734	0.5111	0.0638	0.1305	0.0790	0.0198	0.1515
	RI	90.97%	55.97%	11.16%	1.53%	-0.12%	2.42%	25.73%	65.28%	5.37%	1.26%	1.02%	2.99%
	LightGCN	UNS	0.2085	0.2928	0.3699	<u>0.3745</u>	<u>0.1695</u>	<u>0.5029</u>	0.6588	0.0470	<u>0.1378</u>	<u>0.0874</u>	<u>0.0217</u>
NNCF		0.1600	0.2979	0.3547	0.3576	0.1596	0.4785	<u>0.5473</u>	<u>0.0594</u>	0.1313	0.0825	0.0201	0.1556
DNS		0.2567	0.2728	0.3670	0.3794	0.1712	0.5054	0.6704	0.0453	0.1375	0.0901	0.0220	0.1676
FairStatic		<u>0.0760</u>	0.3505	<u>0.3853</u>	0.3495	0.1570	0.4753	0.5686	0.0591	0.1370	0.0847	0.0211	0.1619
Reg		0.1446	0.3203	0.3744	0.3677	0.1675	0.4998	0.5989	0.0544	0.1356	0.0849	0.0210	0.1599
DPR		0.0768	<u>0.3514</u>	0.3806	0.3614	0.1659	0.4938	0.5900	0.0551	0.1343	0.0842	0.0208	0.1584
FairGAN-2		0.4117	0.1941	0.3300	0.3501	0.1561	0.4633	0.6620	0.0457	0.1351	0.0677	0.0169	0.1229
FairNeg		0.0682	0.3587	0.3865	0.3721	0.1675	0.5020	0.5217	0.0661	0.1383	0.0860	0.0211	0.1598
RI		67.29%	22.51%	4.47%	-0.64%	-1.18%	-0.18%	20.81%	40.64%	0.36%	-1.64%	-2.76%	-4.31%

Table 3: Fairness and recommendation utility of different methods on two backbone models over datasets with a multi-valued item attribute (evaluated on top-20 recommendation results). RI denotes the relative improvement of FairNeg over UNS. We highlight the best results in bold and the second best results with underline.

		ML1M-4						Amazon-4						
Backbone	Method	Recall-Disp.@20	Recall-Min.@20	Recall-Avg.@20	N@20	P@20	R@20	Recall-Disp.@20	Recall-Min.@20	Recall-Avg.@20	N@20	P@20	R@20	
MF	UNS	0.2408	0.1963	0.2931	0.2918	0.1474	0.3765	0.6513	0.0256	0.0921	0.0737	0.0318	0.1020	
	NNCF	0.1603	0.2336	<u>0.3008</u>	0.3006	0.1485	0.3884	<u>0.5240</u>	0.0335	0.0803	0.0599	0.0270	0.0854	
	DNS	0.2350	0.2256	0.3146	0.3203	0.1573	0.4148	0.6596	0.0243	0.0948	<u>0.0751</u>	<u>0.0322</u>	<u>0.1031</u>	
	FairStatic	0.1154	<u>0.2615</u>	0.2931	0.2722	0.1394	0.3624	0.5389	<u>0.0366</u>	0.0979	0.0678	0.0302	0.0953	
	DPR	<u>0.0756</u>	0.2550	0.2878	0.2622	0.1372	0.3549	0.6257	0.0232	0.0725	0.0514	0.0238	0.0708	
	FairGAN-1	0.3012	0.1902	0.2945	0.292	0.1433	0.3762	0.6565	0.0254	<u>0.0990</u>	0.0768	0.0324	0.1013	
	FairNeg	0.0318	0.2998	0.3134	<u>0.3024</u>	<u>0.1508</u>	<u>0.3925</u>	0.5061	0.0439	0.0999	0.0724	0.0321	0.1035	
	RI	86.78%	52.73%	6.90%	3.63%	2.31%	4.25%	22.29%	71.48%	8.47%	-1.76%	0.94%	1.47%	
	LightGCN	UNS	0.1794	0.2455	0.3236	<u>0.3294</u>	<u>0.1603</u>	<u>0.4279</u>	0.6799	0.0247	0.1015	<u>0.0801</u>	<u>0.0344</u>	<u>0.1111</u>
		NNCF	0.1091	0.2571	0.3060	0.3095	0.1498	0.4033	<u>0.5564</u>	0.0342	0.1015	0.0754	0.0327	0.1044
DNS		0.1806	0.2463	<u>0.3259</u>	0.3352	0.1615	0.4374	0.6711	0.0258	0.1053	0.0822	0.0353	0.1125	
FairStatic		0.1022	0.2819	0.3208	0.3156	0.1522	0.4131	0.5679	<u>0.0410</u>	0.1024	0.0770	0.0338	0.1062	
DPR		0.0495	<u>0.2994</u>	0.3207	0.3128	0.1552	0.4084	0.5997	0.0289	0.1017	0.0773	0.0339	0.1075	
FairGAN-2		0.3251	0.1799	0.2999	0.3064	0.1447	0.3980	0.5768	0.0324	0.1003	0.0742	0.0320	0.1004	
FairNeg		0.0394	0.3117	0.3267	0.3232	0.1565	0.4222	0.5062	0.0420	<u>0.1042</u>	0.0792	0.0340	0.1097	
RI		78.04%	26.97%	97.35%	-1.88%	-2.37%	-1.33%	25.55%	70.04%	2.66%	-1.17%	-1.25%	-1.26%	

work for binary group cases, so we only report its performance on **ML1M-2** and **Amazon-2** datasets. FairGAN has a fairness strictness hyperparameter. Thus we set two different values on the two backbone models separately so that the utility is close to other methods, denoted as FairGAN-1 and FairGAN-2. We have the following observations from the results (the evaluation of top-30 recommendation results are presented in the Appendix Section A.3, which also aligns with the observations):

First, compared with FairNeg, most methods provide biased recommendation results regarding the Recall-Disp, Recall-min, and Recall-Avg metrics. Second, compared with the methods that do not explicitly consider group fairness (e.g., NNCF and DNS), fairness-aware methods usually yield better group-wise recall parity. Among them, the adversarial learning-based method usually performs

better than others. It is worth noting that FairStatic shows competitive fairness results, which indicates that optimizing negative sampling distribution with debias considerations also brings substantial fairness improvements.

Third, FairNeg consistently outperforms other methods on different backbone models over four datasets regarding its effectiveness in mitigating group-wise recall performance unfairness. At last, FairNeg can achieve superior fairness while preserving good recommendation utility on the user side. Even though DNS provides the best recommendation utility, it ignores the huge performance disparity among item groups. To conclude, these results validate that our approach FairNeg can effectively improve fairness in group-wise recall performance with minor recommendation utility loss.

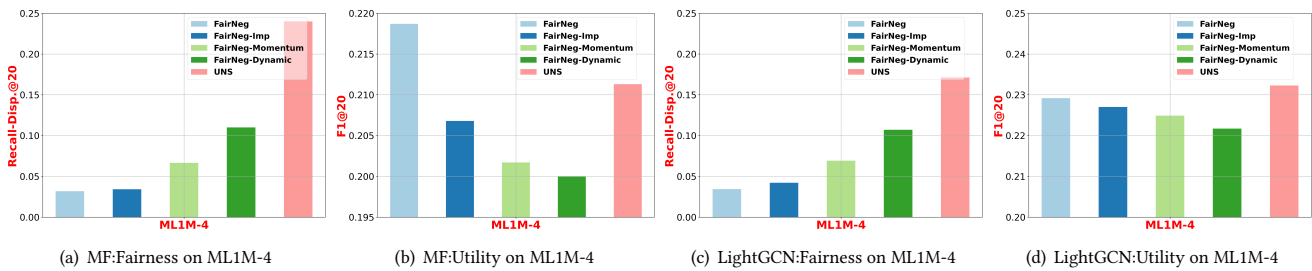


Figure 5: The F1@20 and Recall-Disp@20 of FairNeg and its variants on two backbones over the ML1M-4 dataset.

4.3 Ablation Study

Next, we conduct ablation studies to verify the effectiveness of each component in FairNeg. We compare FairNeg with its variants, where one component are removed (i.e., *Dynamic fair negative sampling* (FairNeg-Dynamic), *Importance-aware sampling* (FairNeg-Imp), and *adaptive momentum update of group NS probability* (FairNeg-Momentum)). The uniform negative sampling (UNS) method served as the baseline. All the methods are evaluated on both MF and LightGCN backbone.

For brevity, we report each variant’s fairness performance (Recall-Disp@20) and the recommendation performance (F1@20) over the ML1M-4 dataset, as presented in Figure 5 (results on Amazon-4 dataset are presented in the Appendix Section A.4).

We have the following findings from the ablation studies. First, the dynamic sampling mechanism (which can be viewed as FairStatic) is vital for fairness improvement and utility-preserving, indicating the importance of adjusting the sampling distribution adaptively. Second, incorporating the importance-aware negative sampling probability distribution is vital for utility preservation, since the informativeness of negative items can be enhanced with such a strategy, further contributing to feature representation learning. Third, removing the momentum mechanism when updating the group-level sampling probability dampens fairness performance, which indicates that the fairness optimization process is unstable and the gradient noise needs to be considered. Fourth, FairNeg achieves the best utility-fairness tradeoff on two backbones, proving that our proposed method’s effectiveness is not limited to a specific model architecture.

5 RELATED WORK

In this section, we briefly review related work, including fair recommendations and negative sampling.

Fair Recommendations. Fair recommendations have drawn tremendous attention in recent years. The fairness notions can be defined from the user side [15, 25, 27] or the item side [16], or both [1], which makes fair recommendation quite challenging. On the item side, many existing works focus on exposure fairness in the ranking results [34, 36, 40], for instance, in [34], a configurable allocation-of-exposure scheme is proposed to overcome richer-get-richer dynamics in the ranking system. However, these methods only study exposure opportunities allocation of items, and the true user preferences are not considered. When items in certain groups have lower recommendation probabilities, given that the items are liked by users, the recommendation model suffers from item under-recommendation

bias. A series of works have been proposed to achieve performance (i.e., recall) parity among different item groups. These methods can be divided into two types: (i) fairness-constrained regularization method [23], where the average score difference of item groups is formed as a regularizer to penalize the group-wise performance disparity; (ii) adversary learning methods [30, 48, 52], where a discriminator classifies the group attribute based on the predicted user-item relevance scores and the recommendation model prevents the discriminator from correctly classifying the groups. Unlike these methods, our approach proposes a fairly adaptive negative sampling technique, which addresses group unfairness from negative sampling perspective.

Negative Sampling. Negative sampling techniques [33] are frequently used in pairwise algorithms [47], and we can group them into static negative sampling and hard negative sampling. The former (i.e., uniform sampling) utilizes a fixed sampling distribution and provides high efficiency. The latter [6, 45, 49] selects the negative instances that are similar to positive instances in the embedding space in a dynamic way. For example, in [35], they adopt adversarial learning to mine hard examples, which brings more information to model training. In [6], they leverage a high-variance based method to choose high-quality negative instances in a reliable way. However, these methods are mainly designed to maximize the overall utility [44], while neglecting the data bias (group-wise) on the item side. Our work makes the first attempt to build the connection between item group fairness optimization and negative sampling.

6 CONCLUSION

Item-oriented group performance fairness is an essential factor for building trustworthy recommender systems [15, 29]. In this work, we propose a novel **Fairly** adaptive **Negative** sampling framework (FairNeg) to alleviate the adverse impact of negative sampling on training recommendation models. Based on the pairwise training paradigm, we introduce the fairness perception module to measure the recall performance disparity and then adjust the group-wise sampling probability with an adaptive momentum mechanism. Furthermore, we introduce the mixup mechanism for combining fairness-aware and importance-related sampling distribution, which aims to jointly consider representation learning and group fairness optimization. Extensive experiments show that our method outperforms state-of-the-art debiasing approaches regarding fairness performance by significant margins and yields better fairness-performance tradeoffs. In the future, we plan to investigate the possibility of leveraging the framework to be compatible with more group fairness metrics in recommendations.

ACKNOWLEDGMENTS

The research described in this paper has been partly supported by NSFC (Project No. 62102335), a General Research Fund from the Hong Kong Research Grants Council (Project No. PolyU 15200021 and PolyU 15207322), and internal research funds from The Hong Kong Polytechnic University (Project No. P0036200, P0042693, and P0043302). This research was also supported by the InnoHK project. Dr. Zitao Liu was partly supported by Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University (2022LSYS003).

REFERENCES

- [1] Robin Burke. 2017. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093* (2017).
- [2] 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. In *arXiv preprint arXiv:2010.03240*.
- [3] Jingfan Chen, Wenqi Fan, Guanghui Zhu, Xiangyu Zhao, Chunfeng Yuan, Qing Li, and Yihua Huang. 2022. Knowledge-enhanced Black-box Attacks for Recommendations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 108–117.
- [4] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *KDD*.
- [5] Ashok Cutkosky and Harsh Mehta. 2020. Momentum improves normalized sgd. In *International conference on machine learning*. PMLR, 2260–2268.
- [6] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *NeurIPS* (2020).
- [7] Wenqi Fan, Tyler Derr, Yao Ma, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep Adversarial Social Recommendation. In *IJCAI*.
- [8] Wenqi Fan, Tyler Derr, Xiangyu Zhao, Yao Ma, Hui Liu, Jianping Wang, Jiliang Tang, and Qing Li. 2021. Attacking black-box recommendations via copying cross-domain user profiles. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1583–1594.
- [9] Wenqi Fan, Wei Jin, Xiaorui Liu, Han Xu, Xianfeng Tang, Suhang Wang, Qing Li, Jiliang Tang, Jianping Wang, and Charu Aggarwal. 2023. Jointly attacking graph neural network and its explanations. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE.
- [10] Wenqi Fan, Qing Li, and Min Cheng. 2018. Deep modeling of social relations for recommendation. In *AAAI*.
- [11] Wenqi Fan, Chengyi Liu, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. 2023. Generative Diffusion Models on Graphs: Methods and Applications. *arXiv preprint arXiv:2302.02591* (2023).
- [12] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph Trend Filtering Networks for Recommendation. In *SIGIR*.
- [13] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A graph neural network framework for social recommendations. *TKDE* (2020).
- [14] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep social collaborative filtering. In *RecSys*.
- [15] Wenqi Fan, Xiangyu Zhao, Xiao Chen, Jingran Su, Jingtong Gao, Lin Wang, Qidong Liu, Yiqi Wang, Han Xu, Lei Chen, et al. 2022. A Comprehensive Survey on Trustworthy Recommender Systems. *arXiv preprint arXiv:2209.10117* (2022).
- [16] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, et al. 2021. Towards long-term fairness in recommendation. In *WSDM*.
- [17] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.
- [18] Elizabeth Gómez. 2020. Characterizing and Mitigating the Impact of Data Imbalance for Stakeholders in Recommender Systems. In *RecSys*.
- [19] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).
- [20] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* (2015).
- [21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.
- [22] 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.
- [23] Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on recommendation independence for a find-good-items task. (2017).
- [24] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. 2020. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems* 33 (2020), 728–740.
- [25] Jurek Leonhardt, Avishek Anand, and Megha Khosla. 2018. User fairness in recommender systems. In *WWW*.
- [26] Jie Li, Yongli Ren, Ke Deng, and et al. 2022. FairGAN: GANs-based Fairness-aware Learning for Recommendations with Implicit Feedback. In *WWW*.
- [27] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented fairness in recommendation. In *WWW*.
- [28] Haifeng Liu, Hongfei Lin, Wenqi Fan, Yuqi Ren, Bo Xu, Xiaokun Zhang, Dongzhen Wen, Nan Zhao, Yuan Lin, and Liang Yang. 2022. Self-supervised learning for fair recommender systems. *Applied Soft Computing* (2022).
- [29] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil K Jain, and Jiliang Tang. 2021. Trustworthy ai: A computational perspective. *arXiv preprint arXiv:2107.06641* (2021).
- [30] Haifeng Liu, Nan Zhao, Xiaokun Zhang, Hongfei Lin, Liang Yang, Bo Xu, Yuan Lin, and Wenqi Fan. 2022. Dual constraints and adversarial learning for fair recommenders. *Knowledge-Based Systems* 239 (2022), 108058.
- [31] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.
- [32] Daniel McNamara. 2019. Equalized odds implies partially equalized outcomes under realistic assumptions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*.
- [33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).
- [34] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*. 429–438.
- [35] Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *WWW*.
- [36] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2020. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *WWW*.
- [37] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*.
- [38] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [39] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. 2021. FairBatch: Batch Selection for Model Fairness. In *ICLR*.
- [40] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *KDD*.
- [41] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *ICML*. PMLR.
- [42] El-Ghazali Talbi. 2013. A taxonomy of metaheuristics for bi-level optimization. In *Metaheuristics for bi-level optimization*. Springer, 1–39.
- [43] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. 2020. Equalization loss for long-tailed object recognition. In *CVPR*.
- [44] Qi Wan, Xiangnan He, Xiang Wang, Jiancan Wu, Wei Guo, and Ruiming Tang. 2022. Cross Pairwise Ranking for Unbiased Item Recommendation. In *WWW*.
- [45] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minmax game for unifying generative and discriminative information retrieval models. In *SIGIR*.
- [46] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. 2022. Disentangled contrastive learning for social recommendation. In *CIKM*.
- [47] Runlong Yu, Yunzhou Zhang, Yuyang Ye, Le Wu, Chao Wang, Qi Liu, and Enhong Chen. 2018. Multiple pairwise ranking with implicit feedback. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1727–1730.
- [48] Guanhua Zhang, Yihua Zhang, Yang Zhang, Wenqi Fan, Qing Li, Sijia Liu, and Shiyu Chang. 2022. Fairness Reprogramming. In *NeurIPS*.
- [49] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*.
- [50] Xiangyu Zhao, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, and Chong Wang. 2021. Autoloss: Automated loss function search in recommendations. In *KDD*.
- [51] Ziwei Zhu, Jianling Wang, and James Caverlee. 2019. Improving top-k recommendation via jointcollaborative autoencoders. In *WWW*.
- [52] Ziwei Zhu, Jianling Wang, James Caverlee, and et al. 2020. Measuring and mitigating item under-recommendation bias in personalized ranking systems. In *SIGIR*.

A APPENDIX

In this appendix, we provide the necessary information for reproducing our insights and experimental results.

A.1 Group-wise Statistics of Datasets

The statistics of the four datasets (i.e., feedback and item number) are illustrated in Table 4. We can observe that the item number distribution is fairly imbalanced on these datasets. Compared with ML1M dataset, Amazon dataset has lower density since the average positive feedback of each item group is much lower.

Table 4: Group statistics in the four datasets.

	Group	#Item	#Feedback	$\frac{\#Feedback}{\#Item}$
ML1M-2	Sci-Fi	211	136,816	648
	Horror	261	57,794	221
Amazon-2	Toy	1,307	30,695	23
	Grocery	1,180	53,961	45
ML1M-4	Sci-Fi	140	80,200	573
	Adventure	129	58,679	455
	Childrens'	162	48,974	302
	Horror	254	54,566	215
Amazon-4	Grocery	708	49,095	69
	Office	864	37,580	43
	Pet	437	14,693	34
	Tool	1,052	28,065	27

A.2 Implementation Details

All the experiments are based on two representative recommender models: MF [38] and LightGCN [22] optimized by Bayesian Personalized Ranking (BPR) loss, which do not explicitly take fairness criterion into considerations in their training paradigm. Our proposed method FairNeg is implemented on the basis of PyTorch and Adam optimization. The baseline models for comparison are implemented on Tensorflow with Adam optimizer. We tune the hyper-parameters

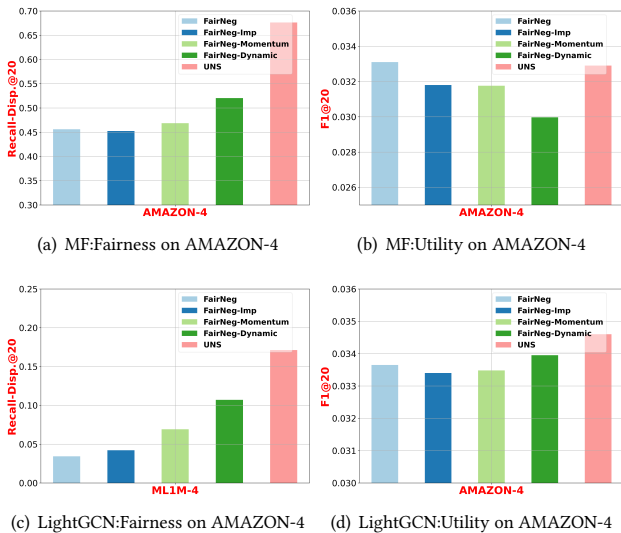
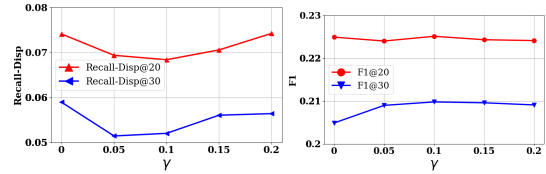


Figure 6: The Fairness/Recommendation utility comparison between FairNeg and its variants over amazon-4 dataset (using MF/LightGCN as backbone).

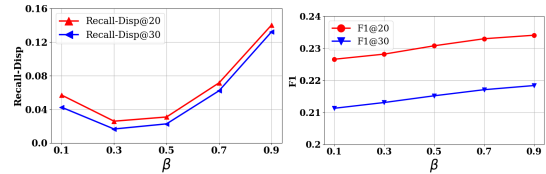
involved in the models with the validation set: the hidden dimensions are searched over [10, 80] and the weight of L_2 regularizer is

searched over {0.01, 0.05, 0.1, 0.5}. Hyper-parameters for our proposed methods include: the momentum coefficient γ is searched over [0, 0.2] with step 0.05 and the utility-fairness trade-off parameter β is searched over [0.1, 0.9] with step 0.1. The learning rate α in the outer optimizer is set as 0.1, temperature τ in the hard-ware probability calculation is set as 0.4. Hyper-parameters defaults for the architecture of recommender system include: the negative sampling ratio set as 1 and mini-batch size set as 1024 for all methods in all experiments. The learning rate for methods using MF as backbone is 0.01 and 0.001 for those using LightGCN. The best models are selected based on the performance on the validation set within 100 epochs.



(a) LightGCN:Fairness on ML1M-4 (b) LightGCN:Utility on ML1M-4

Figure 7: The fairness and utility w.r.t different γ .



(a) LightGCN:Fairness on ML1M-4 (b) LightGCN:Utility on ML1M-4

Figure 8: The fairness and utility w.r.t different β .

A.3 Performance Comparison of Recommender Systems

In this part, We present different methods' performance (user-side recommendation utility and item group fairness) evaluated on **top-30** ranking results.

A.4 Ablation Study

In this part, we show the ablation study results on the Amazon-4 dataset, as presented in Figure 6. The phenomenon is aligned with that on the ML1M-4 dataset. The incorporation of the importance-aware sampling distribution can help to maintain or even improve the overall utility, and the momentum and dynamic mechanism are both essential for item group fairness optimization.

A.5 Hyper-parameter Analysis of FairNeg

In this part, we analyze the influence of two critical hyper-parameters in FairNeg: (i) the momentum parameter γ in Equation 5; (ii) the fairness-utility tradeoff parameter β in Equation 8. For brevity, we report the results on the ML1M-4 dataset using LightGCN as the backbone, and similar patterns can be observed on other datasets/-model backbones.

Impact of γ . To evaluate the influence of the momentum parameter,

Table 5: Fairness and recommendation utility of different methods on two backbone models over datasets with a binary-valued item attribute (evaluated on top-30 recommendation results). RI denotes the relative improvement of FairNeg over UNS. We highlight the best results (in bold) and the second best results with underline.

		ML1M-2						Amazon-2					
Backbone	Method	Recall-Disp.@30	Recall-Min.@30	Recall-Avg.@30	N@30	P@30	R@30	Recall-Disp.@30	Recall-Min.@30	Recall-Avg.@30	N@30	P@30	R@30
MF	UNS	0.2837	0.3142	0.4344	0.3817	<u>0.1364</u>	0.5589	0.6697	0.0530	0.1606	<u>0.0899</u>	<u>0.0169</u>	0.1904
	NNCF	0.1592	0.3344	0.3978	0.3393	0.1193	0.5137	0.5012	<u>0.0796</u>	0.1592	0.0840	0.0161	0.1538
	DNS	0.2822	0.3261	0.4543	0.4097	0.1426	0.5940	0.5875	0.0722	0.1750	0.0972	0.0181	0.2021
	FairStatic	<u>0.0705</u>	<u>0.4242</u>	<u>0.4564</u>	0.3740	0.1323	0.5503	0.5604	0.0705	0.1604	0.0887	0.0164	0.1856
	Reg	0.1928	0.3567	0.4419	0.3495	0.1342	0.5515	0.5305	0.0765	0.1630	0.0871	0.0166	0.1834
	DPR	0.0913	0.4088	0.4499	0.3479	0.1315	0.5286	<u>0.4735</u>	0.0713	0.1275	0.0719	0.0142	0.1633
	FairGAN-1	0.3227	0.2807	0.4145	0.3564	0.1267	0.5136	0.6338	0.0598	0.1632	0.0751	0.0136	0.1478
	FairNeg	0.0221	0.4680	0.4786	<u>0.3880</u>	0.1360	<u>0.5720</u>	0.4706	0.0886	<u>0.1674</u>	0.0888	0.0168	<u>0.1917</u>
	RI	92.21%	48.95%	10.17%	1.65%	-0.29%	2.34%	29.73%	67.17%	4.27%	-1.22%	-0.59%	0.68%
LightGCN	UNS	0.1896	0.3831	0.4727	<u>0.4097</u>	<u>0.1434</u>	<u>0.6043</u>	0.6347	0.0631	0.1728	<u>0.0979</u>	<u>0.0180</u>	<u>0.2048</u>
	NNCF	0.1613	0.3780	0.4507	0.3914	0.1353	0.5753	<u>0.5292</u>	0.0799	0.1697	0.0943	0.0173	0.1985
	DNS	0.2283	0.3620	0.4691	0.4138	0.1444	0.6047	0.6269	0.0661	0.1773	0.1018	0.0185	0.2086
	FairStatic	<u>0.0462</u>	<u>0.4541</u>	<u>0.4866</u>	0.3845	0.1334	0.5770	0.5361	<u>0.0819</u>	0.1766	0.0963	0.0180	0.2029
	Reg	0.1472	0.4069	0.4771	0.4030	0.1424	0.5994	0.5753	0.0745	0.1755	0.0967	0.0180	0.2027
	DPR	0.0945	0.4356	0.4811	0.3961	0.1407	0.5929	0.5864	0.0712	0.1721	0.0960	0.0178	0.2009
	FairGAN-2	0.3749	0.2633	0.4212	0.3802	0.1311	0.5479	0.6220	0.0631	0.1670	0.0746	0.0138	0.1496
	FairNeg	0.0415	0.4704	0.4908	0.4061	0.1420	0.5981	0.4952	0.0893	<u>0.1769</u>	0.0974	0.0179	0.2004
	RI	78.11%	22.79%	3.82%	-0.88%	-0.98%	-1.03%	21.98%	41.52%	2.40%	-0.55%	-0.78%	-2.15%

Table 6: Fairness and recommendation utility of different methods on two backbone models over datasets with a multi-valued item attribute (evaluated on top-30 recommendation results). RI denotes the relative improvement of FairNeg over UNS. We highlight the best results (in bold) and the second best results with underline.

		ML1M-4						Amazon-4					
Backbone	Method	Recall-Disp.@30	Recall-Min.@30	Recall-Avg.@30	N@30	P@30	R@30	Recall-Disp.@30	Recall-Min.@30	Recall-Avg.@30	N@30	P@30	R@30
MF	UNS	0.2296	0.2632	0.3783	0.3215	0.1270	0.4666	0.6330	0.0360	0.1228	0.0854	<u>0.0283</u>	<u>0.1350</u>
	NNCF	0.1479	0.3082	0.3898	0.3321	0.1280	0.4845	0.5053	0.0470	0.1085	0.0707	0.0244	0.1164
	DNS	0.2164	0.2997	0.4058	0.3516	0.1348	0.5087	0.6377	0.0348	0.1237	0.0874	0.0284	0.1390
	FairStatic	0.0920	0.3430	0.3800	0.3012	0.1206	0.4507	<u>0.5035</u>	0.0501	0.1257	0.0781	0.0265	0.1249
	DPR	<u>0.0539</u>	<u>0.3576</u>	0.3767	0.2907	0.1212	0.4505	0.5919	0.0320	0.0956	0.0603	0.0213	0.0961
	FairGAN-1	0.2679	0.2577	0.3768	0.3189	0.1215	0.4594	0.6171	0.0371	<u>0.1264</u>	<u>0.0863</u>	0.0276	0.1286
	FairNeg	0.0253	0.3858	<u>0.4028</u>	<u>0.3333</u>	<u>0.1295</u>	<u>0.4864</u>	0.4816	0.0594	0.1298	0.0832	0.0279	0.1339
	RI	88.98%	46.58%	6.46%	3.67%	1.97%	4.24%	23.92%	65.00%	5.72%	-2.58%	-1.41%	-0.81%
	LightGCN	UNS	0.1664	0.3253	0.4157	<u>0.3609</u>	<u>0.1371</u>	<u>0.5228</u>	0.6357	0.0377	0.1330	<u>0.0925</u>	<u>0.0302</u>
NNCF		0.1033	0.3366	0.3958	0.3414	0.1291	0.4994	0.5326	0.0498	0.1301	0.0871	0.0288	0.1371
DNS		0.1653	0.3261	<u>0.4183</u>	0.3667	0.1379	0.5274	0.6368	0.0377	<u>0.1364</u>	0.0942	0.0310	0.1491
FairStatic		0.0761	0.3720	0.4123	0.3472	0.1309	0.5076	<u>0.5290</u>	0.0551	0.1346	0.0886	0.0295	0.1394
DPR		<u>0.0638</u>	0.3709	0.4121	0.3448	0.1334	0.5061	0.5759	0.0413	0.1318	0.0887	0.0290	0.1397
FairGAN-2		0.2982	0.2408	0.3850	0.3353	0.1234	0.4843	0.5402	0.0454	0.1288	0.0841	0.0276	0.1279
FairNeg		0.0280	0.4020	0.4184	0.3545	0.1338	0.5169	0.4787	0.0556	0.1411	0.0905	0.0295	0.1418
RI		83.17%	23.58%	0.66%	-1.77%	-2.41%	-1.13%	24.70%	46.15%	1.15%	-2.16%	-2.35%	-3.08%

we vary its value in $[0, 0.2]$ with step 0.05. The fairness (Recall-Disp) and recommendation utility (F1) results are evaluated under $K = 20$ and 30 in Figure 7. Here, the importance-aware probability mixup ratio β is set as 0, since we aim to investigate the momentum parameter's influence independently. From the figures, we can find that the fairness performance improves under a proper momentum term (i.e., $\gamma = 0.1$) without hurting the utility.

Impact of β . We also vary the fairness-utility tradeoff parameter in the negative sampling distribution mixup module between $[0.1,$

$0.9]$ with step 0.2 and plot the results in Figure 8. The left figure indicates that with larger β , the fairness can be improved at first but then become worse under $K = 20$ and 30 . We hereby conclude that only properly incorporating importance-aware negative sampling advocates for a better utility-fairness tradeoff. The right figure indicates that by increasing the ratio of importance-aware sampling probability in the mixup procedure, the overall utility can be consistently improved, which attributes to the improvement of the negative samples' informativeness.