FairSync: Ensuring Amortized Group Exposure in Distributed Recommendation Retrieval

Chen Xu
Gaoling School of Artificial Intelligence
Renmin University of China
xc_chen@ruc.edu.cn

Jun Xu[*]
Gaoling School of Artificial Intelligence
Renmin University of China
junxu@ruc.edu.cn

Yiming Ding
Gaoling School of Artificial Intelligence
Renmin University of China
dingym97@ruc.edu.cn

Xiao Zhang
Gaoling School of Artificial Intelligence
Renmin University of China
zhangx89@ruc.edu.cn

Qi Qi
Gaoling School of Artificial Intelligence
Renmin University of China
qi.qi@ruc.edu.cn

ABSTRACT

In pursuit of fairness and balanced development, recommender systems (RS) often prioritize group fairness, ensuring that specific groups maintain a minimum level of exposure over a given period. For example, RS platforms aim to ensure adequate exposure for new providers or specific categories of items according to their needs. Modern industry RS usually adopts a two-stage pipeline: stage-1 (retrieval stage) retrieves hundreds of candidates from millions of items distributed across various servers, and stage-2 (ranking stage) focuses on presenting a small-size but accurate selection from items chosen in stage-1. Existing efforts for ensuring amortized group exposures focus on stage-2, however, stage-1 is also critical for the task. Without a high-quality set of candidates, the stage-2 ranker cannot ensure the required exposure of groups. Previous fairness-aware works designed for stage-2 typically require accessing and traversing all items. In stage-1, however, millions of items are distributively stored in servers, making it infeasible to traverse all of them. How to ensure group exposures in the distributed retrieval process is a challenging question. To address this issue, we introduce a model named FairSync, which transforms the problem into a constrained distributed optimization problem. Specifically, FairSync resolves the issue by moving it to the dual space, where a central node aggregates historical fairness data into a vector and distributes it to all servers. To trade off the efficiency and accuracy, the gradient descent technique is used to periodically update the parameter of the dual vector. The experiment results on two public recommender retrieval datasets showcased that FairSync outperformed all the baselines, achieving the desired minimum level of exposures while maintaining a high level of retrieval accuracy.

CCS CONCEPTS

• **Information systems → Information retrieval**.

KEYWORDS

Distributed Retrieval, Recommender System, Minimum Exposures

ACM Reference Format:
Chen Xu, Jun Xu, Yiming Ding, Xiao Zhang, and Qi Qi. 2024. FairSync: Ensuring Amortized Group Exposure in Distributed Recommendation Retrieval. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3589334.3645413

1 INTRODUCTION

In recent times, the need for fair recommender systems (RS) has gained prominence in industrial requirements [21, 33]. Among these requirements, RS platform has the demands of guaranteeing that specific groups achieve a minimum level of exposure to items within a defined period, aligning with the perspective of amortized group max-min fairness (MMF) [2, 18, 25, 38, 41]. For instance, certain studies propose to ensure minimum item exposures for new providers [4, 28, 38, 39] for attracting providers to join, while others focus on enhancing the visibility of specific item categories [45] for promoting certain festivals. Such "minimum wage policy" [32] significantly contributes to the enhancement of RS, fostering the creation of a more equitable and robust ecosystem.

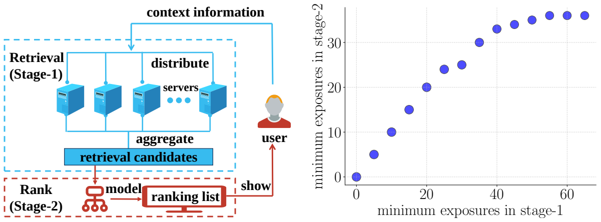
In modern RS, two-stage pipelines have been widely adopted, as shown in Figure 1 (a). The primary objective of stage-1 (retrieval) is to efficiently generate a small set of candidates from millions of items in a distributed manner within milliseconds [5, 8, 20, 23] while stage-2 (ranking) more accurately deals with the candidates selected in stage-1 and generates the final recommendations (usually single-digit items) [38, 45]. Regarding ensuring minimum group exposures in RS, most existing studies [4, 9, 26, 28, 37, 38, 45] primarily concentrated on stage-2.

Existing studies revealed that the fairness task in stage-2 can be compromised if stage-1 fails to retrieve a sufficient number of items [36]. We also conduct a simulation to examine how the minimum exposure in stage-1 affects the minimum exposure in stage-2.

[*]Corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

WWW '24, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0171-9/24/05…$15.00
https://doi.org/10.1145/3589334.3645413



(a) Recommendation workflow (b) Two stage performances

Figure 1: (a) The two-stage pipelines of recommender system, including retrieval (stage-1) and ranking (stage-2). (b) Simulations depicting the changes for the minimum exposures across two stages.

Specifically, we conduct a simulation using Amazon [1] dataset to assess the minimum exposure of groups across two stages. In stage-1, we leverage the YoutubeDNN [8] model and employ a rule-based method to regulate the retrieved exposures of item categories. In stage-2, we implement an oracle ranking model, ensuring the attainment of the highest minimum exposure of groups. The x-axis represents the minimum exposures of item categories in stage-1, while the y-axis corresponds to the exposures of these categories in stage-2. The results reported in Figure 1 (b) indicate that there exists a robust positive correlation between the minimum group exposure of the two stages. In simpler terms, if stage-1 is unable to retrieve required item categories effectively, it will also pose challenges for stage-2 in ensuring the exposure of certain groups.

Though critically important, existing approaches designed for stage-2 [9, 26, 28, 37, 38] cannot be directly applied to stage-1 because they ensure the amortized group exposures by traversing all items and adjusting exposures through aggregating information. During stage-1, however, traversing all items (usually millions of items) is infeasible because (1) these items are distributively stored at different servers, and (2) accessing millions of items causes substantial computational cost. While some heuristic approaches have employed strict rules or adjusted group weights, they still lack the capacity to effectively address the challenge.

In this paper, we introduce a novel model named FairSync, which can ensure the minimum amortized group exposure requirements in the retrieval stage of RS. FairSync converts the problem into a constrained distributed optimization and addresses the problem by transferring it to the dual space. In such space, we aggregate past fairness information into a vector and distribute it to servers. Based on the vector, each server independently conducts the item retrieval. Theoretical analysis shows that, even with local and distributive search, FairSync can still achieve global fairness.

In particular, the RS platform first sets a target to ensure that every group attains a minimum level of exposure. Then we approach the problem by formulating it as a distributed resource allocation problem [1, 38], with the constraint of required exposures. Subsequently, we can transform the constrained optimization problem into an unconstrained dual problem. Then, a constructed dual vector, storing the past fairness information, is combined with user embeddings to form a query vector. Each item, along with its embeddings, is concatenated with the group embedding to form the new item embeddings which will be distributed across servers. After that, each server conducts KNN search for identifying candidate items within milliseconds by using the dense retrieval [42] architecture. Finally, the outcomes are aggregated into a set of candidate items for stage-2. As for the learning procedure, we employ the gradient descent method [17] to update the parameters of dual vector periodically to trade-off the efficiency and effectiveness.

We summarize the major contributions of this paper as follows:

(1) We emphasize the critical importance of incorporating the assurance of minimum exposure for specific groups into the distributed stage-1 (retrieval) of RS.

(2) We introduce a model named FairSync which is tailored to meet the distributed, efficient, and online demands of the prevalent dense retrieval architecture in stage-1 of RS.

(3) The experimental results on two publicly available large-scale recommendation datasets clearly demonstrate that FairSync outperforms the baseline models, attaining the desired minimum level of exposures while preserving a high level of retrieval accuracy.

2 RELATED WORK

Fairness has emerged as a prominent research theme within recommender systems. In this realm, two predominant aspects are often explored: individual fairness [22, 24], which concentrates on equitable treatment for individuals, and group fairness, which categorizes items into various groups such as providers [4, 26, 28, 37–39], and item categories [11, 30, 36]. In group fairness, there are usually two criteria. One is egalitarian proposes [11, 24, 26, 36, 37], which aims to equalize the outcome of different groups, another is Rawl's principle [18], which aims to improve the utility of worst-off groups [4, 28, 38, 39]. In real application of RS, amortized fairness [3, 4, 28, 38, 39] is more realistic, which achieves fairness over a period of time, rather than enforcing it strictly on a single ranking list. In our research, we mainly focused on the amortized group max-min fairness, which is used to support new providers or enhance the visibility of specific item categories.

In RS, there are many methods proposed to alleviate amortized group MMF. FairRec [28] and its extension FairRec+ [4] proposed an offline recommender model to guarantee equal frequency for all items in a series of ranking lists. Yang and Ai [40] proposed a marginal optimizing approach to conduct amortized MMF in the learning-to-rank process. TFROM [37] and CP-Fair [26] proposed a Linear Programming (LP)-based method to ensure the group fairness, see also [2, 9, 41]. P-MMF [38], LTP-MMF [39] proposed an online mirror gradient descent to improve worst-off provider's exposures in the dual space. Nonetheless, all of these proposals have been introduced within the context of stage-2 scenarios, making them impractical for application in stage-1 due to their substantial computational overhead.

In large RS, the significance of stage-1 (retrieval) cannot be overstated, as the performance of stage-2 is heavily reliant on it [5, 19, 20, 36, 43]. There are also some works that proposed inspiring approaches to solve fairness issues in stage-1. Wang and Joachims [36] proposed an uncertainty quantification approach to

[1] http://jmcauley.ucsd.edu/data/amazon/

control the threshold of each retrieval channel in one retrieval process. Hao et al. [13], Rastegarpanah et al. [29] proposed a fairness-related matrix factorization method to adjust the weight of the retrieval model. In resource allocation, Balseiro et al. [1], Cheung et al. [7] proposed a mirror-descent method to solve in the dual space. However, these methods either fall into addressing amortized group max-min fairness well or are unsuitable for implementation within the retrieval systems that require distributive, efficient, and online capabilities.

3 PROBLEM FORMULATION

In RS, let $\mathcal{U}, \mathcal{I}$ be the set of users and items, and each item $i \in \mathcal{I}$ is associated with a unique group $g \in \mathcal{G}$. The set of items associated with a specific group g is denoted as $\mathcal{I}_g$. When a specific user $u \in \mathcal{U}$ accesses the retrieval system, the system will retrieve items from distributed servers and aggregate them into a list of candidate items with a predefined size of K, denoted by $L_K(u) \in \mathcal{I}^K$, which is then prepared for stage-2 for detailed ranking.

In real-world applications, the users arrive at the RS sequentially. Assume that at time t, user u_t arrives. The RS aims to ensure that the exposure of a specific group g remains at or exceeds a threshold of m_g throughout the entire time horizon from $t = 1$ to T, all the while optimizing to retain enough relevant items within a single candidate retrieval list. At the same time, we require an online solution, where at time step t, the RS responds to a request from user u_t by providing a candidate list without waiting for input from a second user u_{t+1}. An online retrieval algorithm h produces a real-time decision candidates $L_K(u_t)$ based on the current user u_t and the previous history $\mathcal{H}_{t-1} = \{u_s, L_K(u_s)\}_{s=1}^{t-1}$:

$$L_K(u_t) = h(u_t \mid \mathcal{H}_{t-1}, \mathcal{M}),$$

where $\mathcal{M} = \{m_g | g \in \mathcal{G}\}$ is the factor set by the platform.

State-of-the-art recommender retrieval models [5, 19, 20, 43] usually employ the distributive dense retrieval architecture, wherein an item i is represented as an embedding $\boldsymbol{e_i} \in R^d$ using complex neural networks, such as transformers [35]. These embeddings are indexed on each server $S_n, n \in [1, 2, \cdots, M]$ in a distributed manner, with $d \in N^+$ being the predefined dimension and M is the server number. For the user u, a simple network is employed to represent them as an embedding $\boldsymbol{e}_u \in R^d$, typically utilizing their historical browsing information in state-of-the-art systems. The user-item relevance score $r_{u,i}$ is calculated as the distance between $\boldsymbol{e}_i$ and $\boldsymbol{e}_u$ locally in each server S_n. The retrieval model's objective is to identify candidate items whose embeddings $\boldsymbol{e}_i$ are in close proximity to the embedding of the user $\boldsymbol{e}_u$, i.e., finding the highest possible relevance scores $r_{u,i}$ in the candidate list for the stage-2 ranking process.

Generally, the RS will establish the offline index [16] for items to efficiently search the desired ones from each server. Previous research [9, 28, 38] focus on ensuring amortized group exposure $\mathcal{M}$ by traversing all items i and their corresponding groups g, where $i \in \mathcal{I}_g$. However, in a distributed dense retrieval architecture, these methods are no longer suitable.

4 OUR APPROACH

In this section, we will introduce our approach FairSync.

4.1 Distributed Dense Retrieval Architecture

In the mainstream recommender retrieval architectures, the primary objective is to identify items whose embeddings $\boldsymbol{e}_i$, are in close proximity to the embedding of the user $\boldsymbol{e}_u$ distributively. Formally, the problem can write as:

$$L_K(u) = \mathop{\arg\min}_{L\subset\{1,2,\ldots,|\mathcal{I}|\},|L|=k} \sum_{i\in S_n,\forall n} d(\boldsymbol{e}_u, \boldsymbol{e}_i), \tag{1}$$

where L is the set of indices of the K nearest neighbors, $d(\boldsymbol{e}_u, \boldsymbol{e}_i)$ is the distance between embedding $\boldsymbol{e}_u, \boldsymbol{e}_i$, i-th the commonly used distance metric being the dot-product locally on each server, i.e. $d(\boldsymbol{e}_u, \boldsymbol{e}_i) = -\boldsymbol{e}_u^\top \boldsymbol{e}_i$, and the $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$ are calculated by a complex model, such as Deep Neural Network [8], Recurrent Neural Network [15], Capsule Network [20], i.e.

$$\boldsymbol{e}_u = m^u(u), \quad \boldsymbol{e}_i = m^i(i),$$

where $m^u(\cdot)$ and $m^i(\cdot)$ are two embedding extraction networks.

Typically, the item embeddings $\boldsymbol{e}_i$ are pre-calculated and distributively indexed on servers [16], whereas the user embedding $\boldsymbol{e}_u$ requires online inference using complex recommendation models, see [5, 8, 20, 43]. In real application [5], Equation (1) is computed by performing KNN search in the embedding space efficiently.

4.2 Dual Space of Retrieval

After the platform gives the minimum exposure requirement for each group, i.e., requiring the exposure of a specific group g to remain at or exceed a threshold of m_g throughout the entire time horizon from $t = 1$ to T. Therefore, we write the equation (1) as a distributed resource allocation problem:

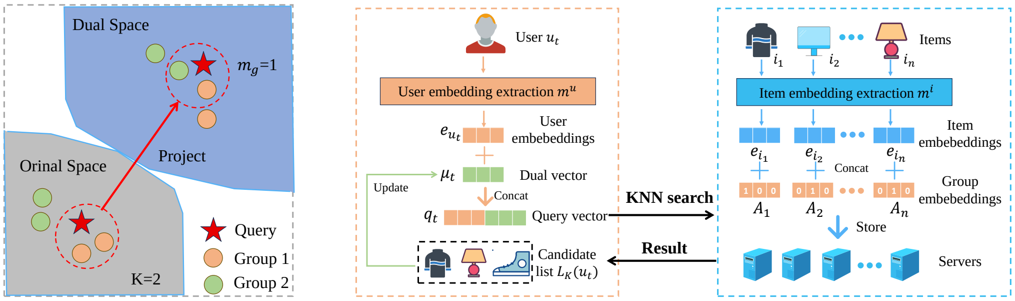
$$\begin{aligned}
\max_{x_{u_t,i}} \quad & \sum_{t=1}^{T} \sum_{i\in S_n,\forall n} x_{u_t,i} r_{u_t,i} \\
\text{s.t.} \quad & \sum_{i\in S_n,\forall n} x_{u_t,i} = K, \quad \forall t \in [1, 2, \ldots, T] \\
& r_{u_t,i} = -d(\boldsymbol{e}_{u_t}, \boldsymbol{e}_i) \\
& e_g = \sum_{t=1}^{T} \sum_{i\in\mathcal{I}_g} x_{u_t,i}, \quad \forall g \in \mathcal{G} \\
& e_g \geq m_g, \quad \forall g \in \mathcal{G} \\
& x_{u_t,i} \in \{0, 1\}, \forall t \in [1, 2, \ldots, T], i \in \mathcal{I}
\end{aligned} \tag{2}$$

where e_g can be seen as the total number of exposed items of group g, accumulated over the period 1 to T, $x_{u_t,i} \in \{0, 1\}$ is the decision vector for user u_t. Specifically, for each item i, $x_{u_t,i} = 1$ if it is added to the candidate list $L_K(u_t)$, otherwise $x_{u_t,i} = 0$.

Theorem 1. *The dual problem objective W^{Dual} of Equation (2) can be write as*

$$\min_{\boldsymbol{\mu}} \left[\sum_{t=1}^{T} \sum_{k=1}^{K} (r_{u_t,i} - \boldsymbol{A}\boldsymbol{\mu})_{[k]} + \sum_{g\in\mathcal{G}} m_g \boldsymbol{\mu}_g + \max_g \{\boldsymbol{\mu}_g\}(TK - \sum_{g\in\mathcal{G}} m_g) \right], \tag{3}$$

where we can have a dual variable $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{G}|}$, $\mathbf{A} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{G}|}$ is the item-group adjacent matrix, and $A_{ig} = 1$ indicates item $i \in \mathcal{I}_g$, and 0 otherwise. Moreover, the dual problem is a strong-dual problem, that is the optimal value of Equation (3) is the same as Equation (2).



(a) Illustrative Example (b) Online Retrieval Process (c) Offline Item Servers

Figure 2: FairSync Framework. Sub-figure (a) illustrates an example to show the intuitive example of how FairSync works. Sub-figure (b) illustrates that the online retrieval process when user u_t arrives, while sub-figure (c) depicts the offline item embeddings in the dual space construction process.

Remark 1 (Distributed solution in dual space). *After the transformation of the original problem (Equation (1)) into its dual form (Equation (1)), we can convert the problem into an unconstrained optimization problem, simplifying the optimization process significantly. We can also observe that different items are independent of each other. Therefore, the problem can be effectively solved in a distributed manner. The value $\boldsymbol{\mu}$ can be regarded as the accumulated exposure information and be distributed to each server during the retrieval process.*

Remark 2 (Small computational costs). *The original problem (Equation (1)) is computationally intensive due to its nature as a constraint integral linear problem, and it involves a vast variable space of size $T \times |\mathcal{I}|$ given that the retrieval process may encompass millions of items. However, in the dual problem (2), we observe that the variable size has been significantly reduced to $|\mathcal{G}| \ll T \times |\mathcal{I}|$, and thanks to the sparsity of $\mathbf{A}$, the computation of $\mathbf{A}\boldsymbol{\mu}$ is highly efficient. This operation serves to project the variable $\boldsymbol{\mu}$ from the group space into the item spaces.*

The detailed proof can be seen in the Appendix A.2.

4.3 FairSync Algorithm

Figure 2 shows the framework of the FairSync. FairSync will retrieve items from the transformed dual space. Next, we will illustrate the inference phase and online learning phase, respectively.

4.3.1 Inference phase. we will provide illustrative examples to demonstrate how FairSync works and present a detailed inference workflow of FairSync.

Firstly, Figure 2 (a) provides an illustrative example to demonstrate the functioning of the FairSync algorithm in an intuitive manner. In this example, we simplify the problem by retrieving two items from a corpus containing four items (depicted as circles in the figure), each assigned to different groups, represented by distinct colors in the figure. A user u_t arrives at the recommender system, and this user is represented as the red pentagram. The system's requirement is to ensure that there is at least one exposure of each group. From the original space, the user and items are represented as the embedding $\boldsymbol{e}_u, \boldsymbol{e}_i$ in Section 4.1. In such space, the retrieval system will find the top-2 closest items, i.e. orange items to users. However, in the dual space, FairSync will project the user and item embeddings to different points, while ensuring the minimum exposure constraint is satisfied. In the dual space, the dense retrieval system can efficiently locate the distributed items that meet the requirements and simultaneously maintain retrieval accuracy.

Algorithm 1: FairSync Algorithm

```
Input: User arriving order $\{u_t\}_{t=1}^{T}$, item corpus $\mathcal{I}$, candidate size
    $K$, batch size $B$, optimizer Opt with learning rate $\eta$, trained
    user item embedding network $m^u(\cdot), m^i(\cdot)$ item-group
    adjacent matrix $\mathbf{A}$, minimum group exposure requirement
    $\{m_g\}_{g\in\mathcal{G}}$.
Output: The candidate lists for every user $\{L_K(u_t)\}_{t=1}^{T}$
Calculate items embeddings $\{\boldsymbol{e}_i = m^i(i), \forall i \in \mathcal{I}\}$
Re-construct and distributively index the item embeddings
    $\{\boldsymbol{h}_i = \boldsymbol{e}_i \| \boldsymbol{A}_i, \forall i \in \mathcal{I}\}$.
Initialize update count $b = 0$.
Initialize the gradient buffer $\mathcal{B}_s = \{\}$.
for $t = 1, \cdots, T$ do
    Initialize dual solution $\boldsymbol{\mu} = 0$
    User $u_t$ arrives
    Calculate user embedding $\boldsymbol{e}_{u_t} = m^u(u_t)$
    Re-construct query embeddings $\boldsymbol{q}_{u_t} = \boldsymbol{e}_{u_t} \| \boldsymbol{\mu}$
    // KNN Retrieval: (Equation (4))
    $L_K(u_t) = \arg\min_{S\subset\{1,2,\ldots,|\mathcal{I}|\},|S|=k} \sum_{i\in S} d^{\text{Dual}}(\boldsymbol{q}_{u_t}, \boldsymbol{h}_i)$,
    Compute the sub-gradient $\boldsymbol{s}$ utilizing the Equation (6)
    Store the sub-gradient $\boldsymbol{s}$ into $\mathcal{B}_s$
    Update count $c = c + 1$
    if $c = B$ // Update per B users then
        $\boldsymbol{u} = \text{Opt}(\boldsymbol{\mu}, \sum_{s\in\mathcal{B}_s} \boldsymbol{s})$
        Initialize update count $b = 0$.
        Initialize the gradient buffer $\mathcal{B}_s = \{\}$
    end if
end for
```

Formally, from Theorem 1, we can observe that the distance between the user and item in the dual space transforms to:

$$d^{\text{Dual}} = d(\boldsymbol{e}_u, \boldsymbol{e}_i) + \boldsymbol{\mu}_g, \quad i \in \mathcal{I}_g. \tag{4}$$

Therefore, to better adapt to the dense retrieval architecture

Table 1: Statistics of the datasets.

Dataset	#User	#Item	#Group	#Interaction
Amazon-Book	459,133	313,966	165	8,898,041
Taobao	976,779	1,708,530	1246	85,384,110

discussed in Section 4.1, we reconstruct user u_t embeddings to $\boldsymbol{q}_{u_t}$ and embedding of item i to $\boldsymbol{h}_i$, where $\boldsymbol{q}_{u_t}$ and $\boldsymbol{h}_i$ are defined as follows:

$$\boldsymbol{q}_{u_t} = \boldsymbol{e}_{u_t} \| -\boldsymbol{\mu}_t, \quad \boldsymbol{h}_i = \boldsymbol{e}_i \| \boldsymbol{A}_i, \tag{5}$$

where $\|$ denotes the concat operator between two vectors and $\boldsymbol{A}_i$ denotes the i-th column vector of adjacent matrix in Theorem 1. Therefore, we have $d^{\text{Dual}} = -\boldsymbol{q}_{u_t}^\top \boldsymbol{h}_i$.

Figure 2 (b) illustrates the inference phase of FairSync in a more visualized way. Firstly, a user u_t arrives, then the user embedding extraction m^u module (any retrieval model) will extract the user embedding $\boldsymbol{e}_{u_t}$. Then, at time t, we have a dual vector $\boldsymbol{\mu}_t$ to form the query vector $\boldsymbol{q}_{u_t}$ (Equation (5)). Then we will utilize the vector $\boldsymbol{q}_{u_t}$ to utilize k-nearest neighbors (KNN) search on the distributively indexed item embeddings $\{\boldsymbol{h}_i, \forall i \in \mathcal{I}\}$ in the dual space to retrieve a corresponding list of candidate items (Figure 2 (c)).

4.3.2 Online learning phase. In the online learning phase, we aim to update the dual vector $\boldsymbol{\mu}_t$ once in a while.

Specifically, we can see that the sub-gradient $\boldsymbol{s} \in \mathbb{R}^{|\mathcal{G}|}$, $\boldsymbol{s} \in \partial W^{\text{Dual}}/\partial \boldsymbol{\mu}_t$ in Equation (3) at time t satisfies:

$$s_g = \begin{cases} m_g + \sum_{i \in L_K(u_t)} I(i \in \mathcal{I}_g), & \text{if } g \neq \hat{g} \\ \sum_{i \in L_K(u_t)} I(i \in \mathcal{I}_g) + (TK - \sum_{g \neq \hat{g}} m_g), & \text{else,} \end{cases} \tag{6}$$

where $I(\cdot)$ denotes the indicator function and $\hat{g} = \arg\max_{g \in \mathcal{G}} \boldsymbol{\mu}_g$.

Based on the assumption that user comes to the system randomly [38], we can utilize the sub-gradient $\boldsymbol{s}$ to update $\boldsymbol{\mu}_t$. In real applications, however, updating $\boldsymbol{\mu}_t$ at every time step t is challenged by a large number of asynchronous update operations on different servers, and when the update frequency is too high, it can lead to excessively long recall times, thereby impacting the user experience. Therefore, to trade off the efficiency and effectiveness, we will update the dual vector $\boldsymbol{\mu}$ each B steps.

Specifically, we will store the sub-gradient $\boldsymbol{s}$ of each step into a gradient buffer $\mathcal{B}_s$. For each B steps, we will utilized any optimizer Opt (in this paper, we utilized the well-performing Adam [17]) to update $\boldsymbol{\mu}$ utilizing the averaged gradient in the buffer, i.e. $\boldsymbol{\mu} = \text{Opt}(\boldsymbol{\mu}, \sum_{s \in \mathcal{B}_s} \boldsymbol{s})$.

The detailed FairSync algorithm is shown in Algorithm 1.

4.3.3 Discussion. Our algorithm FairSync can be applied with any other distributed retrieval architecture. We will illustrate the methods in detail. In retrieval architectures, most of the work commonly incorporates ANN algorithms, as elucidated in the paper. There are also tree-based indexing models such as TDM [44] and similar indexing approaches. As we solely map the user embedding and item embedding to the dual space outlined in Equation (5), our approach allows for adapting with various distributed retrieval architectures. This is possible as other methods can easily apply our approach by indexing the dual item embedding using their preferred indexing techniques.

5 EXPERIMENT

We conducted experiments to demonstrate the effectiveness of the proposed FairSync. The source code and experiments have been shared at github[2].

5.1 Experimental settings

5.1.1 Datasets. Following the practice in Cen et al. [5], the experiments are conducted on one commonly used publicly available retrieval datasets and one billion-scale industrial dataset, including:

Amazon-Book[3]: The subsets (book domains) of Amazon [14] Product dataset. The item grouping relies on the field "categories". Each training sample is truncated at length 20. As a pre-processing step, we consider groups with fewer than 50 items as a single group, which we name the "infrequent group".

Taobao[4]: collected about 1 million user behaviors data based on Taobao's recommender systems [44] during November 25 to December 03, 2017. The item grouping relies on the field "category ID". Each training sample is truncated at length 50. As a pre-processing step, we consider groups with fewer than 200 items as a single group, which we name the "infrequent group".

The statistics of the two datasets are shown in Table 1.

5.1.2 Evaluation. Firstly, following the common practice [5, 19, 20], we train the embedding extraction network $m^i(\cdot), m^u(\cdot)$. We sort all the interactions in the dataset based on their timestamps and utilize the initial 80% of the interactions as the training data for $m^i(\cdot), m^u(\cdot)$ training. The remaining 20% of interactions were split into two equal parts, with each 10% portion serving as the validation and test data, respectively, for evaluation.

As for the evaluation metrics, the performances of the models were evaluated from two aspects: retrieval accuracy, and the minimum group exposure satisfaction (i.e. performance of fairness). Let T be the test set length and $\hat{\mathcal{I}}_u$ be the set of items for user u.

For the retrieval accuracy, following [5, 6], we utilize

- **Recall**:

$$\text{Recall@N} = \frac{1}{T}\sum_{t=1}^{T} \frac{|L_K(u_t) \cap \hat{\mathcal{I}}_{u_t}|}{\hat{\mathcal{I}}_{u_t}}.$$

- **Hit Rate.** The HR Rate (HR) is a metric that quantifies the percentage of recommended items that include at least one item that the user has previously interacted with [5, 6].

$$\text{HR@N} = \frac{1}{T}\sum_{t=1}^{T} I(|L_K(u_t) \cap \hat{\mathcal{I}}_{u_t}| > 0).$$

- **Normalized Discounted Cumulative Gain.** Normalized Discounted Cumulative Gain (NDCG) is a metric that factors in the positions of correctly recommended items, providing a measure that accounts for the item's relevance and its position in the recommendation list [5].

$$\text{NDCG@N} = \frac{1}{T}\sum_{t=1}^{T} \sum_{i \in L_K(u_t)} \frac{I(i \in \hat{I}_{u_t})}{\log_2(\text{pos}(i, L_K(u_t)))} / Z_t,$$

where $\text{pos}(i, L_K(u_t))$ is the sorting position of item i in the

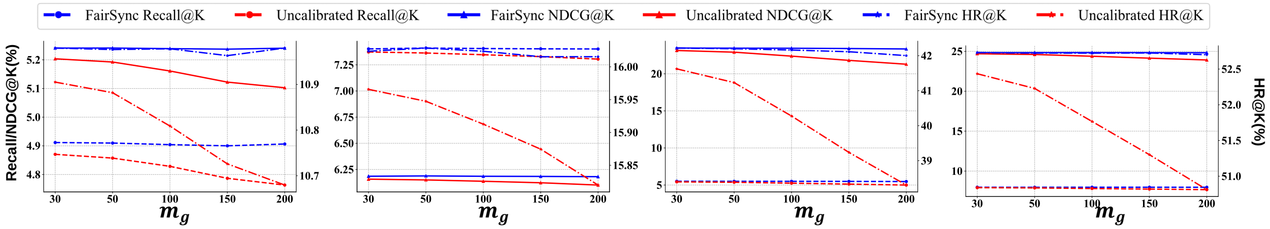
[2] https://github.com/XuChen0427/FairSync

[3] http://jmcauley.ucsd.edu/data/amazon/

[4] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649&userId=1

Table 2: Performance comparisons between ours and the baselines on Amazon book subset and Taobao. Our objective is to guarantee that each group possesses a minimum of 200 exposures to fulfill the ESP metric. The $*$ means the improvements over the baseline that can guarantee minimum exposure baselines (K-neighbor and Uncalibrated) are statistically significant (t-tests and p-value < 0.05). The bold number indicates that the accuracy value exceeds that of all the baselines. All the numbers in the table are percentage numbers with "%" omitted.

Base model	Fairness model	Amazon-Book dataset								Taobao dataset							
		top-20				top-50				top-20				top-50			
		Recall	NDCG	HR	ESP	Recall	NDCG	HR	ESP	Recall	NDCG	HR	ESP	Recall	NDCG	HR	ESP
youtubeDNN	regularized-fair	4.52	4.61	10.13	53.94	7.11	5.64	15.55	81.21	3.29	14.85	28.89	58.27	4.97	16.56	39.31	82.83
	IPW	4.55	4.64	10.19	45.45	7.16	5.68	15.66	73.94	3.29	14.85	28.89	57.78	4.97	16.56	39.31	82.66
	K-neighbor	0.09	0.14	0.29	100.00	0.14	0.17	0.41	100.00	0.15	0.87	1.73	100.00	0.24	1.00	2.51	100.00
	Uncalibrated	4.44	4.53	9.96	100.00	7.08	5.62	15.51	100.00	2.99	13.46	26.18	100.00	4.79	15.95	37.87	100.00
	FairSync(ours)	**4.55***	**4.64***	**10.19***	100.00	**7.16***	**5.69***	**15.68***	100.00	**3.29***	14.77*	28.74*	100.00	**4.99***	**16.56***	**39.32***	100.00
GRU4REC	regularized-fair	3.95	4.01	8.70	46.67	6.35	4.94	13.63	77.58	4.73	18.84	35.63	64.93	6.95	20.43	45.99	83.63
	IPW	3.97	4.04	8.76	38.79	6.38	4.97	13.70	63.03	4.73	18.84	35.64	64.69	6.95	20.43	45.99	83.55
	K-neighbor	0.09	0.13	0.26	100.00	0.14	0.15	0.41	100.00	0.17	0.79	1.54	100.00	0.24	0.92	2.19	100.00
	Uncalibrated	3.90	3.94	8.58	100.00	6.32	4.91	13.55	100.00	4.29	17.08	32.26	100.00	6.69	19.65	44.25	100.00
	FairSync(ours)	**3.98***	**4.04***	**8.77***	100.00	6.37*	**4.97***	13.68*	100.00	**4.74***	18.79*	35.52*	100.00	**6.96***	**20.54***	**46.01***	100.00
MIND	regularized-fair	6.64	6.58	13.70	41.82	9.64	7.66	19.46	63.64	4.62	18.98	36.15	62.28	6.96	20.70	47.34	79.21
	IPW	6.62	6.56	13.67	38.18	9.63	7.63	19.42	58.79	4.62	18.98	36.15	62.28	6.96	20.70	47.34	78.97
	K-neighbor	0.10	0.16	0.32	100.00	0.15	0.18	0.40	100.00	0.17	0.94	1.80	100.00	0.26	1.12	2.60	100.00
	Uncalibrated	6.45	6.39	13.33	100.00	9.52	7.54	19.20	100.00	4.20	17.23	32.80	100.00	6.69	19.93	45.58	100.00
	FairSync(ours)	6.60*	**6.60***	13.65*	100.00	**9.65***	**7.69***	**19.48***	100.00	4.57*	18.82*	35.86*	100.00	**6.98***	**20.76***	**47.38***	100.00
ComiRec-DR	regularized-fair	4.92	5.26	10.99	37.58	7.40	6.20	16.03	61.21	5.51	23.49	42.25	63.24	7.98	24.85	52.77	80.26
	IPW	4.91	5.24	10.97	33.33	7.41	6.18	16.03	55.15	5.51	23.49	42.25	63.24	7.98	24.85	52.76	80.26
	K-neighbor	0.09	0.14	0.25	100.00	0.14	0.16	0.37	100.00	0.19	1.01	1.85	100.00	0.28	1.18	2.60	100.00
	Uncalibrated	4.76	5.10	10.68	100.00	7.30	6.10	15.82	100.00	4.99	21.29	38.30	100.00	7.67	23.92	50.81	100.00
	FairSync(ours)	**4.92***	**5.28***	**11.0***	100.00	**7.42***	**6.20***	**16.08***	100.00	5.47*	23.35*	42.20*	100.00	**8.07***	**24.93***	**52.80***	100.00
ComiRec-SA	regularized-fair	5.23	3.78	10.83	49.70	8.09	4.93	16.47	75.76	5.49	23.77	41.61	63.88	7.76	24.98	51.28	80.10
	IPW	5.25	3.79	10.85	44.85	8.10	4.93	16.46	70.91	5.49	23.77	41.62	63.80	7.76	24.99	51.28	80.10
	K-neighbor	0.11	0.14	0.29	100.00	0.15	0.75	1.92	100.00	0.17	0.90	1.61	100.00	0.25	1.10	2.39	100.00
	Uncalibrated	5.12	3.70	10.59	100.00	8.01	4.88	16.30	100.00	4.97	21.53	37.65	100.00	7.47	24.06	49.36	100.00
	FairSync(ours)	**5.26***	**3.80***	10.81*	100.00	**8.12***	**4.93***	**16.47***	100.00	5.45*	23.66*	41.36*	100.00	**7.76***	**24.99***	**51.33***	100.00



(a) Amazon-book K=20 (b) Amazon-book K=50 (c) Taobao K=20 (d) Taobao K=50

Figure 3: The accuracy curve (Recall, NDCG, and HR) of FairSync (ours) and the best baseline Uncalibrated under different minimum exposure threshold m_g. The experiments were conducted based on the best retrieval base model ComiRec-DR.

list $L_K(u_t)$, starting from 1 to K and Z_t represents a normalization constant that denotes the ideal discounted cumulative gain (IDCG@N), which signifies the highest achievable value for the numerator in the metric at time t.

For the minimum group exposure satisfaction, we apply:

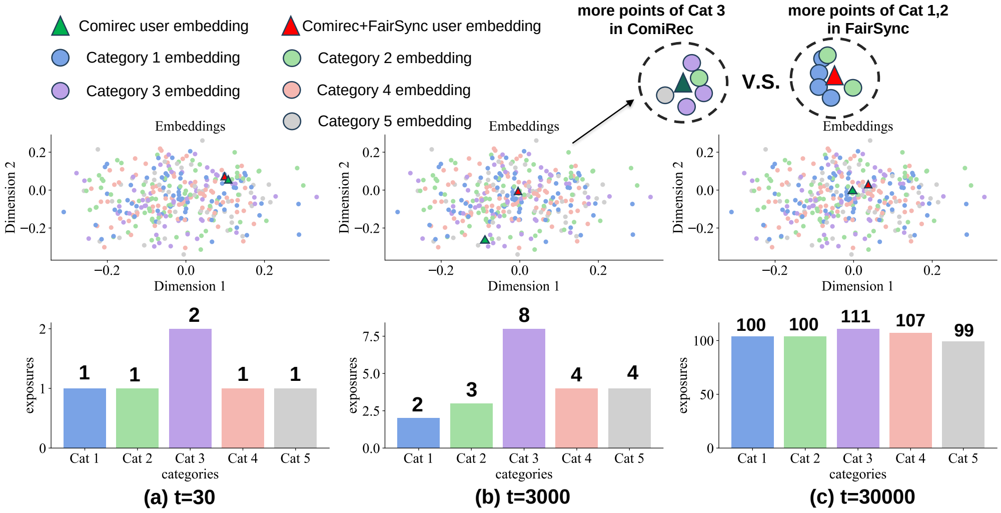
- **Enough Satisfaction Groups.** Enough satisfaction groups (ESP) aims to estimate whether each candidate generation policy selects enough items that satisfy the minimum group exposure requirement, similar to the enough relevant items (ER) metric in [36]:

$$\text{ESP} = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} I\left(\left[\sum_{t=1}^{T}\sum_{i\in L_K(u_t)} I(i\in\mathcal{I}_g)\right] > m_g\right).$$

5.1.3 Baselines and Base Models. In this section, we mainly introduce the controllable retrieval baselines and base models used for extraction user and item embeddings.

For the distributed retrieval baseline, we mainly choose four heuristic methods: **regularized-fair** [38]: at each time step t, a regularized-based dual variable to reduce the exposure gaps between all items and the worst-off item. **IPW** [36]: selected the group exposure as the item's inverse propensity weighted (IPW) during the retrieval process. However, the two aforementioned baselines fail to ensure the necessary minimum exposures for groups, as employing different trade-off coefficients λ reveals an inability to comprehensively fulfill the fairness constraint.

The next baselines are the two heuristic methods used to ensure that the required minimum exposures of groups are guaranteed in the retrieval process. K**-neighbor** [28]: at each time step t, Only the items on each server associated with the top-K group, having the lowest cumulative exposure, are retrieved. **Uncalibrated** [36]: each step t only chooses the items whose group does not satisfy the required exposures. For a fair comparison, we also retrieve them using the KNN search method.

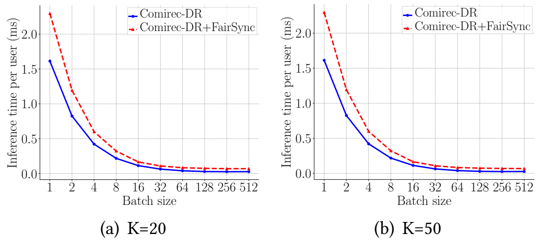


(a) t=30

(b) t=3000

(c) t=30000

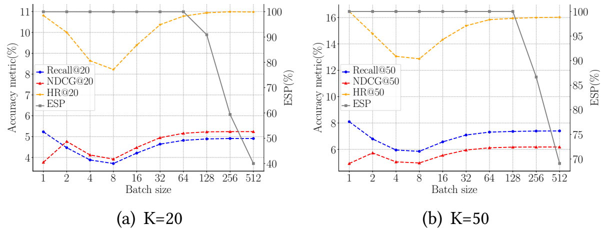
Figure 4: The three sub-figures in the first row illustrate the t-SNE visualization item embeddings and user embeddings of Comirec-DR and our model FairSync using Comirec-DR as the base model under different time t. The three sub-figures in the final row depict the category exposures under different time steps t. The experiment was conducted on the Amazon-book dataset with retrieval number $K = 50$.



(a) K=20

(b) K=50

Figure 5: Inference time per user w.r.t. w.r.t. batch size B.



(a) K=20

(b) K=50

Figure 6: Recall, NDCG, HR and ESP curves of FairSync under different top-K ranking w.r.t. batch size B.

For the retrieval base models, we utilize: **Youtube DNN** [8]: the most commonly used retrieval models in industrial recommender systems; **GRU4Rec** [15]: utilized the recurrent neural network (RNN) to model the user sequential behaviors in the retrieval process; **MIND** [20]: aimed to model user's diverse interests by designing a multi-interest extractor layer based on the capsule routing mechanism [12]; **ComiRec-SA** [5]: the recent state-of-the-art retrieval models, which captured user diverse interests by the self-attention mechanism. **ComiRec-DR** [5]: the variant of ComiRec-SA, which used the dynamic routing method to model user's sequential behaviors.

5.1.4 Implementation details. As for the hyper-parameters in all models, the learning rate η was tuned among $[1e-2, 1e-4]$, and the batch size for updating dual vector B was tuned among $[1, 512]$. For training the base retrieval model, we utilize the best parameters reported in the original papers of the models. We implement FairSync with the most common faiss [16] KNN-search package. The gradient descent package used Pytorch [27] to apply the autogradient. The experiments were conducted under a server with a single NVIDIA GeForce RTX 3090.

5.2 Experimental Results on Full Datasets

Firstly, we conduct experiments to show the performance of FairSync and other baselines under the same minimum exposure requirement ($m_g = 200, \forall g \in \mathcal{G}$) across all retrieval base models. Table 2 presents the experimental outcomes for our FairSync model and the baseline methods across all datasets, while ensuring that each group maintains a minimum of 200 exposures as a requirement. To make fair comparisons, all the baselines were tuned their hyperparameters to obtain the best performance under our settings.

Based on the reported findings, it becomes evident that our model FairSync effectively fulfills the requirement of each group maintaining a minimum of 200 exposures (i.e., ESP=100%). Furthermore, FairSync significantly outperforms the baseline techniques intended for guaranteeing minimum exposure (K-neighbor and Uncalibrated) across all datasets and various base retrieval models, encompassing different top-K retrieval numbers, as reflected in accuracy metrics including Recall, NDCG, and HR. Simultaneously, FairSync exhibits accuracy performance that is comparable with other fairness baselines (regularized-fair and IPW), even though these methods do not strictly ensure the required minimum exposure of groups. The experiments conclusively demonstrate that FairSync effectively guarantees the minimum exposure requirement without significantly compromising the accuracy of the retrieval process.

Secondly, we conduct experiments to demonstrate the performance of FairSync and the best baseline (Uncalibrated) under varying minimum exposure requirements under the best retrieval base model ComiRec-DR. Figure 3 reports the accuracy (Recall, NDCG, and HR) curve of our model FairSync and the best baseline Uncalibrated under different minimum exposure threshold $m_g \in [10, 200], \forall g \in \mathcal{G}$. Both FairSync and Uncalibrated are able to satisfy the minimum exposure requirements.

From the curves presented in Figure 3 (a-d), it is evident that our model FairSync consistently outperforms Uncalibrated with a large margin across various accuracy metrics, datasets, and retrieval numbers ($K = 20, 50$). The experiment demonstrates that our model FairSync consistently exhibits better accuracy when ensuring the minimum exposure requirements of different groups.

5.3 Experiment analysis

We also conduct experiments to analyze FairSync on Amazon-book dataset under the best retreival base model ComiRec-DR. For other analysis, please see Appendix C.

5.3.1 Visualization of embeddings under original and dual space. In this section, we aim to visualize and illustrate the effective dual projection of FairSync (with ComiRec-DR as the base model, i.e. ComiRec+FairSync) by randomly sampling 5 item categories and setting the requirement $m_g = 200, \forall g \in \mathcal{G}$. Figure 4 utilizes t-SNE [34] to visually represent user and item embeddings $\boldsymbol{e}_{u_t}$ and $\boldsymbol{e}_i$ in the original space (sub-figures in first rows), as well as user and item embeddings $\boldsymbol{q}_{u_t}$ and $\boldsymbol{h}_i$ in the dual space (sub-figures in second rows), across various time steps t. We also show the category exposures under different time steps t (sub-figures in third rows). Note that ComiRec-DR is a multi-interest retrieval model [5, 20], where we set four user embeddings generated to represent different user interests per time step t.

Figure 4 (a) illustrates that at the initial retrieval process ($t = 30$), the exposure levels for various categories (as depicted in the third column's bar plots) are nearly equalized. Such equalized exposure, in turn, leads to FairSync's reconstructed embeddings in the dual space (ComiRec-DR+FairSync embeddings) closely mirroring the patterns of the original embeddings (ComiRec-DR embeddings) to maintain retrieval accuracy.

Figure 4 (b, c) illustrates the intermediary and ending stage ($t = 3000, 30000$) of stage-1, during which category 3 dominates in exposure levels, whereas the other categories exhibit a lower level of exposure. In the original space (ComiRec-DR embeddings), it is evident that the user embeddings are closely aligned with the embeddings of category 3. However, in the dual space (ComiRec-DR+FairSync embeddings), the user embeddings are in closer proximity to lower categories (1,2), thereby ensuring that other categories meet the minimum exposure requirements.

The experiment clearly demonstrated that throughout the retrieval process, our model FairSync dynamically adjusts the user embedding's position based on category exposure, enhancing retrieval accuracy while maintaining the minimum exposure requirement.

5.3.2 Ablation study on batch size. In this section, we aim to conduct experiments to show the performance and inference time influenced by different online batch size B, since B controls the dual vector $\boldsymbol{\mu}$'s updating frequency. Figure 5 and Figure 6 depict the variations in inference time and performance, respectively, with respect to the batch size $B \in [1, 512]$.

Firstly, Figure 5 illustrates the online inference time per user w.r.t. batch size under different retrieval numbers K. From the displayed curve, it is evident that when the batch size is smaller ($B \le 8$), FairSync still demands approximately $[0.2, 1]$ ms more time in comparison to the base model. When the batch size is relatively large ($B > 8$), the inference times of both FairSync and the base model are comparable, typically remaining below 0.25 ms. This satisfies the inference time requirements for industrial applications.

Secondly, Figure 6 illustrates that accuracy (Recall, NDCG and HR) curve and ESP ($m_g = 200$) curve w.r.t. batch size under different retrieval number K. Based on the depicted curve, it is apparent that the retrieval accuracy curve decreases as the batch size varies within the range $B \in [1, 8]$, whereas for batch sizes within the range $B \in [8, 512]$, the accuracy curve exhibits an increase. It is also worth noting that the minimum exposure requirement is no longer satisfied as the batch size increases beyond $B > 64$.

Therefore, we observe that the online batch size B is a tradeoff co-efficient for performance and inference time. In real-world applications, we must carefully control the online batch size B, as larger values can reduce inference time but may result in poorer performance, while smaller values can have the opposite effect.

6 CONCLUSION

This paper emphasizes the importance of considering amortized fairness in stage-1 of RS. Then, we propose a novel retrieval model called FairSync that aims to maintain accuracy while ensuring the minimum exposure for specific groups in the distributed retrieval process. In FairSync, we transform the problem into a constrained distributed optimization problem and resolve the issue in the dual space of the problem in a distributed manner. Extensive experiments conducted on two large-scale datasets consistently showcased FairSync's superior performance over baseline models across various retrieval base models. Importantly, FairSync manages to maintain minimal computational costs in real-world applications.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (No. 62376275), Beijing Key Laboratory of Big Data Management and Analysis Methods, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, funds for building world-class universities (disciplines) of Renmin University of China. Supported by the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China (22XNKJ07), and the Outstanding Innovative Talents Cultivation Funded Programs 2024 of Renmin University of China.

REFERENCES

[1] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. 2021. Regularized online allocation problems: Fairness and beyond. In *International Conference on Machine Learning*. PMLR, 630–639.

[2] Omer Ben-Porat and Rotem Torkan. 2023. Learning with Exposure Constraints in Recommendation Systems. In *Proceedings of the ACM Web Conference 2023*. 3456–3466.

[3] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*. 405–414.

[4] Arpita Biswas, Gourab K Patro, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2021. Toward Fair Recommendation in Two-sided Platforms. *ACM Transactions on the Web (TWEB)* 16, 2 (2021), 1–34.

[5] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[6] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.

[7] Wang Chi Cheung, Guodong Lyu, Chung-Piaw Teo, and Hai Wang. 2020. Online planning with offline simulation. *Available at SSRN 3709882* (2020).

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[9] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2021. Two-sided fairness in rankings via Lorenz dominance. *Advances in Neural Information Processing Systems* 34 (2021), 8596–8608.

[10] Ky Fan. 1953. Minimax theorems. *Proceedings of the National Academy of Sciences* 39, 1 (1953), 42–47.

[11] 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 *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 445–453.

[12] Taeyoung Hahn, Myeongjang Pyeon, and Gunhee Kim. 2019. Self-routing capsule networks. *Advances in neural information processing systems* 32 (2019).

[13] Qianxiu Hao, Qianqian Xu, Zhiyong Yang, and Qingming Huang. 2021. Pareto Optimality for Fairness-Constrained Collaborative Filtering. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 5619–5627. https://doi.org/10.1145/3474085.3475706

[14] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[16] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

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

[18] Julian Lamont. 2017. *Distributive justice*. Routledge.

[19] Beibei Li, Beihong Jin, Jiageng Song, Yisong Yu, Yiyuan Zheng, and Wei Zhou. 2022. Improving Micro-video Recommendation via Contrastive Multiple Interests. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2377–2381.

[20] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.

[21] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2022. Fairness in recommendation: A survey. *arXiv preprint arXiv:2205.13619* (2022).

[22] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards personalized fairness based on causal notion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1054–1063.

[23] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. 2020. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*. 463–473.

[24] Mirko Marras, Ludovico Boratto, Guilherme Ramos, and Gianni Fenu. 2022. Equality of learning opportunity via individual fairness in personalized recommendations. *International Journal of Artificial Intelligence in Education* 32, 3 (2022), 636–684.

[25] David Matsumoto and Linda Juang. 2016. *Culture and psychology*. Cengage Learning.

[26] Mohammadmehdi Naghiaei, Hossein A Rahmani, and Yashar Deldjoo. 2022. Cpfair: Personalized consumer and producer fairness re-ranking for recommender systems. *arXiv preprint arXiv:2204.08085* (2022).

[27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[28] 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 *Proceedings of The Web Conference 2020*. 1194–1204.

[29] Bashir Rastegarpanah, Krishna P Gummadi, and Mark Crovella. 2019. Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 231–239.

[30] Dimitris Sacharidis. 2019. Top-n group recommendations with fairness. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing*. 1663–1670.

[31] Harvey M Salkin and Cornelis A De Kluyver. 1975. The knapsack problem: a survey. *Naval Research Logistics Quarterly* 22, 1 (1975), 127–144.

[32] Asep Suryahadi, Wenefrida Widyanti, Daniel Perwira, and Sudarno Sumarto. 2003. Minimum wage policy and its impact on employment in the urban formal sector. *Bulletin of Indonesian economic studies* 39, 1 (2003), 29–50.

[33] Chun-Hua Tsai and Peter Brusilovsky. 2021. The effects of controllability and explainability in a social recommender system. *User Modeling and User-Adapted Interaction* 31 (2021), 591–627.

[34] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[36] Lequn Wang and Thorsten Joachims. 2023. Uncertainty Quantification for Fairness in Two-Stage Recommender Systems. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 940–948.

[37] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. 2021. Tfrom: A two-sided fairness-aware recommendation model for both customers and providers. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1013–1022.

[38] Chen Xu, Sirui Chen, Jun Xu, Weiran Shen, Xiao Zhang, Gang Wang, and Zhenhua Dong. 2023. P-MMF: Provider Max-min Fairness Re-ranking in Recommender System. In *Proceedings of the ACM Web Conference 2023*. 3701–3711.

[39] Chen Xu, Xiaopeng Ye, Jun Xu, Xiao Zhang, Weiran Shen, and Ji-Rong Wen. 2023. LTP-MMF: Towards Long-term Provider Max-min Fairness Under Recommendation Feedback Loops. *arXiv preprint arXiv:2308.05902* (2023).

[40] Tao Yang and Qingyao Ai. 2021. Maximizing marginal fairness for dynamic learning to rank. In *Proceedings of the Web Conference 2021*. 137–145.

[41] Tao Yang, Zhichao Xu, and Qingyao Ai. 2022. Effective Exposure Amortizing for Fair Top-k Recommendation. *arXiv preprint arXiv:2204.03046* (2022).

[42] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1503–1512.

[43] Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-seng Chua, and Fei Wu. 2022. Re4: Learning to re-contrast, re-attend, re-construct for multi-interest recommendation. In *Proceedings of the ACM Web Conference 2022*. 2216–2226.

[44] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1079–1088.

[45] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 449–458.

Algorithm 2: regularized-fair Algorithm

```
Input: User arriving order $\{u_t\}_{t=1}^{T}$, item corpus $\mathcal{I}$, candidate size
    $K$, batch size $B$, trained user item embedding network
    $m^u(\cdot), m^i(\cdot)$ item-group adjacent matrix $\mathbf{A}$, trade-off
    coefficient $\lambda$.
Output: The candidate lists for every user $\{L_K(u_t)\}_{t=1}^{T}$
Calculate items embeddings $\{\boldsymbol{e}_i = m^i(i), \forall i \in \mathcal{I}\}$
Re-construct and distributively index the item embeddings
    $\{\boldsymbol{h}_i = \boldsymbol{e}_i \| \boldsymbol{A}_i, \forall i \in \mathcal{I}\}$ utilizing the Equation (5).
for $t = 1, \cdots, T$ do
    User $u_t$ arrives
    Calculate user embedding $\boldsymbol{e}_{u_t} = m^u(u_t)$
    Compute $\boldsymbol{\mu}_r = \lambda[\mathbf{A}(e - (\min_{g\in\mathcal{G}} e_g \mathbf{1}^\top)]$
    Re-construct query embeddings $\boldsymbol{q}_{u_t} = \boldsymbol{e}_{u_t} \| - \boldsymbol{\mu}_r$.
    // KNN Retrieval: (Equation (4))
    $L_K(u_t) = \arg\min_{S\subset\{1,2,\ldots,|\mathcal{I}|\},|S|=k} \sum_{i\in S} d^{\mathrm{Dual}}(\boldsymbol{q}_{u_t}, \boldsymbol{h}_i)$,
end for
```

A APPENDIX

A.1 Lemma 1

Firstly, we prove a lemma before we start the proof of Theorem 1.

Lemma 1. *Let $\boldsymbol{a}_{[i]}$ denotes the i-th largest element of $\boldsymbol{a}$. Considering the function with the $\boldsymbol{x} \in \mathbb{R}^N$ as the input,Top-K($\boldsymbol{x}$) $= \sum_{k=1}^{K} \boldsymbol{x}_{[k]}$. We demonstrate that the function $f(\boldsymbol{x})$ exhibits concavity w.r.t. $\boldsymbol{x}$.*

Proof. By the definition, for any $0 \le \lambda \le 1$ we have

$$\begin{aligned}\text{Top-K}(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) &= \sum_{k=1}^{K} (\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y})_{[k]} \\ &\le \lambda \sum_{k=1}^{K} \boldsymbol{x}_{[k]} + (1-\lambda) \sum_{k=1}^{K} \boldsymbol{y}_{[k]} \\ &= \lambda\text{Top-K}(\boldsymbol{x}) + (1-\lambda)\text{Top-K}(\boldsymbol{y}),\end{aligned}$$

that is the sum of the first k elements of two vectors added together is less than the sum of the first k elements of the two vectors individually added. □

A.2 Proof of Theorem 1

Proof. We can utilize the Lagrangian condition [1, 38] to decompose the relation between e_g and $\sum_{t=1}^{T}\sum_{i\in\mathcal{I}_g} x_{u_t,i}$:

$$\begin{aligned} W \le W^{Dual} = \max_{\boldsymbol{x}_{u_t,i}\in\mathcal{X}} \min_{\boldsymbol{\mu}} \quad & \sum_{t=1}^{T}\sum_{i=1}^{|\mathcal{I}|} \left[(r_{u_t,i} - \boldsymbol{A}_i^\top \boldsymbol{\mu}) \boldsymbol{x}_{u_t,i} + \sum_{g\in\mathcal{G}} \boldsymbol{\mu}_g e_g \right] \\ \text{s.t.} \quad & e_g \ge m_g, \quad \forall g \in \mathcal{G} \\ & \sum_{g\in\mathcal{G}} e_g = TK. \end{aligned}$$

Where $\mathcal{X} = \{\boldsymbol{x}_{u_t,i} | \boldsymbol{x}_{u_t,i} = \{0,1\}, \sum_{i\in\mathcal{I}} \boldsymbol{x}_{u_t,i} = K\}$ is the feasible region of $\boldsymbol{x}_{u_t,i}$ and the $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{G}|}$ is the dual vector. Note that the condition $\sum_{g\in\mathcal{G}} e_g = TK$ should be satisfied before the dual transformation. This precaution is necessary because without this condition e_g could easily diverge, therefore, we introduce an upper bound for e_g.

Let's consider the following program:

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \boldsymbol{a}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{x} = K, 0 \le \boldsymbol{x} \le \mathbf{1}, \end{aligned} \tag{7}$$

This problem is a well-studied knapsack problem [31], whose optimal objective should be $\sum_{i=1}^{K} \boldsymbol{a}_{[i]}$. The equation tells us that only the top K items that user u_t have the highest preference for every group p will be recommended for every user.

Thus, we can easily observe that the objective W of the target about $x_{t,i}$ is a top-K function in lemma 1 and from lemma 1, we can observe that W is concave with respect to x and convex with respect to the variable $\boldsymbol{\mu}$. From the minimax theorem [10], we can re-write the equation as:

$$\begin{aligned} W = \min_{\boldsymbol{\mu}} \max_{e_g} \quad & \sum_{t=1}^{T}\sum_{k=1}^{K} \left[(r_{u_t,i} - \boldsymbol{A}_i^\top \boldsymbol{\mu})_{[k]} + \sum_{g\in\mathcal{G}} \boldsymbol{\mu}_g e_g \right] \\ \text{s.t.} \quad & e_g \ge m_g, \quad \forall g \in \mathcal{G} \\ & \sum_{g\in\mathcal{G}} e_g = TK \end{aligned}, \tag{8}$$

Now, consider the following problem:

$$\begin{aligned} L = \max_{\boldsymbol{e}} \quad & \sum_{g\in\mathcal{G}} \boldsymbol{\mu}_g e_g \\ \text{s.t.} \quad & \sum_{g\in\mathcal{G}} \boldsymbol{e}_g = TK, \boldsymbol{e}_g \ge m_g, \quad \forall g \in \mathcal{G}, \end{aligned}$$

which is a well-studied knapsack problem [31], with the optimal solution

$$\sum_{g\in\mathcal{G}} m_g \boldsymbol{\mu}_g + \max_{g\in\mathcal{G}}\{\boldsymbol{\mu}_g\}(TK - \sum_{g\in\mathcal{G}} m_g).$$

Finally, we can take the optimal solution into Equation (8), we get W^{Dual} as

$$min_{\mu} \left[\sum_{t=1}^{T}\sum_{k=1}^{K} (r_{u_t,i} - \boldsymbol{A}\boldsymbol{\mu})_{[k]} + \sum_{g\in\mathcal{G}} m_g \boldsymbol{\mu}_g + \max_{g}\{\boldsymbol{\mu}_g\}(TK - \sum_{g\in\mathcal{G}} m_g) \right].$$

□

B REGULARIZED-FAIR ALGORITHM

In this section, we propose a heuristic method for distributed approach for improving the worst-off group exposures in retrieval process, aligning with the concept amortized max-min fairness [9, 38], named regularized-fair. Similar with the dual form of FairSync, it introduced a dual variable $\boldsymbol{\mu}_r$ that measures the exposure gaps between the target group and the worst-groups. The detailed algorithm is shown in Algorithm 2.

C ADDITIONAL EXPERIMENTAL ANALYSIS

C.1 FairSync performance under extreme scenario

In the context of amortized fairness in our settings, an extreme scenario might occur where there is a group for which all users express a unanimous dislike for the items associated with that group. So we conducted a toy experiment to show how our method FairSync will perform under such extreme cases. Let's assume there

Table 3: The performance of FairSync and ComiRec-DR under an extreme case. All the numbers in the table are percentage numbers with % omitted.

Model	Recall	ESP
ComiRec-DR	1.0	50.0
ComiRec-DR+FairSync(ours)	0.96	100.0

Table 4: The performance of FairSync and other baseline performances when m_g is randomly sampled from [1,200] for every $g \in \mathcal{G}$ under $K = 20$. All the numbers in the table are percentage numbers with % omitted.

Model	Recall	NDCG	HR	ESP
K-neighbor	0.09	0.14	0.25	100.0
Uncalibrated	4.82	5.15	10.8	100.0
FairSync(ours)	**4.91**	**5.24**	**11.0**	100.0

are two groups: g_1 and g_2, and each group has 5 items. Meanwhile, $10,000$ users are accessing the retrieval system, and each user is conducting retrievals for a set of $K = 5$ items. Each user's embeddings will exhibit proximity (distance ranging from 0 to 0.4) to the item embeddings belonging to category g_1, while maintaining a distance (ranging from 0.4 to 1.0) from the item embeddings associated with category g_2 . Every group will require 2000 exposures (m_1=m_2=2000). Our experiment results are shown in Table 3.

From the experiments, we can see the recall will be 0.96 (decrease 4% = $\frac{m_2}{KT}$ compared to methods without fairness constraints), and the fairness constraints will also be satisfiable (g_1 will get 48000 exposures while g_2 will get 2000 exposures). Certainly, incorporating constraints incurs a marginal reduction in recall without adversely impacting latency. However, as detailed in the subsequent question, our method supports query-level constraints. This enables the platform to strategically decrease the requirement m_2 on g_2 to achieve higher accuracy, presenting a trade-off decision for the platform.

C.2 Performance on group-level requirements.

In our main experiments, we have chosen to maintain a constant value for m_g across all groups g. We can make different user groups to select different m_g to achieve the group-level constraints. We conduct an experiment to randomly sample from [1,200] for every group g to show our effectiveness for $K = 20$.

From Table 4, the reported results indicate that our method, FairSync, continues to effectively address group-level constraints, demonstrating superior retrieval accuracy compared to other baselines such as K-neighbor and Uncalibrated.