Item-side Fairness of Large Language Model-based Recommendation System

Meng Jiang
University of Science and Technology of China
Hefei, China
jiangm@mail.ustc.edu.cn

Keqin Bao
University of Science and Technology of China
Hefei, China
baokq@mail.ustc.edu.cn

Jizhi Zhang
University of Science and Technology of China
Hefei, China
cdzhangjizhi@mail.ustc.edu.cn

Wenjie Wang*
National University of Singapore
Singapore
wenjiewang96@gmail.com

Zhengyi Yang
University of Science and Technology of China
Hefei, China
yangzhy@mail.ustc.edu.cn

Fuli Feng
University of Science and Technology of China
Hefei, China
fulifeng93@gmail.com

Xiangnan He*
MoE Key Lab of BIPC
University of Science and Technology of China
xiangnanhe@gmail.com

ABSTRACT

Recommendation systems for Web content distribution intricately connect to the information access and exposure opportunities for vulnerable populations. The emergence of Large Language Models-based Recommendation System (LRS) may introduce additional societal challenges to recommendation systems due to the inherent biases in Large Language Models (LLMs). From the perspective of item-side fairness, there remains a lack of comprehensive investigation into the item-side fairness of LRS given the unique characteristics of LRS compared to conventional recommendation systems. To bridge this gap, this study examines the property of LRS with respect to item-side fairness and reveals the influencing factors of both historical users' interactions and inherent semantic biases of LLMs, shedding light on the need to extend conventional item-side fairness methods for LRS. Towards this goal, we develop a concise and effective framework called IFairLRS to enhance the item-side fairness of an LRS. IFairLRS covers the main stages of building an LRS with specifically adapted strategies to calibrate the recommendations of LRS. We utilize IFairLRS to fine-tune LLaMA, a representative LLM, on *MovieLens* and *Steam* datasets, and observe significant item-side fairness improvements. The code can be found in https://github.com/JiangM-C/IFairLRS.git.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation, Large Language Model, Item-side Fairness

ACM Reference Format:
Meng Jiang, Keqin Bao, Jizhi Zhang, Wenjie Wang, Zhengyi Yang, Fuli Feng, and Xiangnan He. 2024. Item-side Fairness of Large Language Model-based Recommendation System. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3589334.3648158

1 INTRODUCTION

Recommendation systems play a pivotal role in the distribution of diverse Web content, encompassing news reports [45], social posts [12], micro-videos [61], and a range of descriptions like clothing [32], cuisine [30], pharmaceuticals [14], and job [35]. Recommendation systems are intricately linked to societal concerns such as equitable information access and exposure opportunities for vulnerable populations. Recently, Large Language Models (LLMs) [5, 44] have gained prominence as a key in recommendation systems due to their exceptional content comprehension capabilities [1, 2, 57]. Nevertheless, the integration of LLMs may introduce societal challenges to recommendation systems, primarily stemming from the inherent biases in LLM training datasets [33, 44]. Therefore, there is a pressing need to investigate the trustworthiness of LLM-based Recommendation Systems (LRS).

Item-side Fairness (IF) [24, 48] is a critical aspect of trustworthy recommendations, aiming to provide fair exposure opportunities for different item groups. IF is widely used to ensure the rights and profit of item producers such as the opportunity to seek appropriate candidates of micro-businesses in job recommendation [6]. Additionally, IF can also be applied to different topics, facilitating effective dissemination of content (*e.g.,* news reports) on societal issues like environmental sustainability and climate action. Furthermore, IF can elevate the visibility of content related to vulnerable populations, ensuring their interests and demands receive adequate attention from society. Despite its significance, there is a lack of comprehensive research investigating IF in the context of LLM-based recommendation systems.

LLM-based recommendation systems [3, 52, 57] exhibit unique characteristics compared to conventional recommendation systems [25, 51]. These include their reliance on semantic clues to infer user preferences, the use of instructions to describe the recommendation task, and the generation of recommendations instead of

[*] Corresponding authors.

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

ACM ISBN 979-8-4007-0171-9/24/05…$15.00
https://doi.org/10.1145/3589334.3648158

relying solely on discriminative predictions. Consequently, previous findings regarding item-side fairness in conventional methods may not hold true for LLM-based systems. Therefore, it is crucial to examine the property of LRS with respect to item-side fairness. This study specifically investigates two key questions:

- **RQ1:** How does LRS perform in terms of item-side fairness compared to traditional recommendation models?
- **RQ2:** What is the root cause of the fairness issue in the LRS?

To answer these questions, we conduct exploratory experiments on two public datasets under a sequential recommendation setting as per recent LRS studies [1, 2]. We compare LLM-based methods, represented by BIGRec [1], with SASRec [17], representative of conventional sequential recommendation methods in terms of item-side fairness. Our findings indicate that LRS is notably influenced by the popularity factor, as BIGRec consistently recommends more popular items compared to SASRec. Additionally, BIGRec exhibits biases towards certain item groups in specific genres (*e.g.,* crime), suggesting the impact of inherent semantic biases within the LLM since item genre is related to the textual description of items (inputs of LRS). In summary, the imbalanced distribution of historical interactions for LRS training and the inherent semantic bias are both contributing factors to the unfairness observed in LRS. Consequently, it is crucial to adapt conventional methods to enhance item fairness by addressing these factors in LRS.

To achieve this target, we develop IFairLRS, a concise and effective framework that enhances the item-side fairness of an LRS. IFairLRS calibrates the recommendations of LRS to meet the IF requirements by considering the two stages of building an LRS: in-learning, and post-learning. For each stage, IFairLRS incorporates specific strategies adapted from IF methods for conventional recommendation models [40, 58]. In the in-learning stage, IFairLRS reweights training samples based on the bias observed between the distribution of target items and historical interactions. In the post-learning stage, IFairLRS can rerank the recommendations of LRS by incorporating a punishment term regarding unfairness. Experiments on real-world datasets demonstrate the effectiveness of our framework in enhancing item-side fairness *w.r.t.* both historical interactions and semantic biases. Our codes and data will be released upon acceptance.

In conclusion, our contributions are as follows:

- We reveal the unfairness issue of LRS and the unique characteristics of LRS in terms of item-side fairness.
- We develop a concise and effective framework for the item-side fairness of LRS with strategies for different training phases.
- We conduct extensive experiments on real-world datasets, validating the effectiveness of the proposed framework with an in-depth analysis of the pros and cons of different strategies.

2 RELATED WORK

In this section, we review the recent work on item-side fairness in recommendation and LLM-based recommendation systems.

2.1 Item-side Fairness in Recommendation

The item-side fairness considers whether the items are treated fairly by the recommendation system [24, 48], and can be categorized into two primary classes: individual fairness and group fairness. Individual fairness requests each individual item be treated similarly [4, 22, 41, 62] and group fairness demands each predefined item group be treated equally [10, 11, 31, 38, 47]. Our work falls under the category of group fairness. Different from the previous work, we examine the IF of LRS.

To the best of our knowledge, there is only one work LLMRank [15] associated with IF in LRS. However, LLMRank merely qualitatively observes the correlation between the frequency of item appearances in the recommendation results of LRS and the popularity of the item, without further systematic analysis like the quantitative study. Different from LLMRank, we design multiple metrics to quantitatively analysis the IF in LRS from popularity and genre two perspectives and systematically propose methods to improve the IF of LRS.

2.2 LLM-based Recommendation System

LLMs, which consist of billions of parameters, have reformulated the paradigm of natural language processing [19, 50]. Different from traditional language models like BERT [7], GPT-2 [36], BART [21], and so on, LLMs show much stronger natural language understanding and generation ability since LLMs have billions or more parameters, train on more data, and have more elaborate network structure [33, 34, 49, 60]. Among them, LLaMA [44] has garnered widespread acceptance in academia for its open-source nature. Thus, we adopt LLaMA as the LLM backbone of our research.

Astonished by the miracle exhibited by LLMs, there has sparked a trend in contemplating how to utilize the power of LLMs in the field of recommendation system [8, 23, 28, 42, 46, 52, 59]. Current research on LRS follows a general pipeline: translating recommendation data into natural language input and then utilizing LLMs to generate recommendation results in a natural language form [15, 26, 29, 56, 57]. However, due to limitations such as the lack of recommendation data during the pre-training phase of LLMs, directly using LLMs for recommendation can only achieve suboptimal performance, making it necessary to tune LLMs on the recommendation data to unleash their recommendation capabilities [1, 2, 54]. BIGRec [1] has the potential to serve as a representative of the LRS approach since BIGRec encapsulates the fundamental elements of LRS and numerous methods can be conceptualized as further extensions grounded in the BIGRec paradigm. Different from previous LRS user-side fairness evaluation work FaiRLLM [56], we take an exploratory approach from the item side, addressing a crucial aspect that is currently lacking in fairness research in LRS.

3 PRELIMINARY

In this section, we first elaborate on the evaluation of item-side fairness in LLM-based recommendation systems. Afterward, we introduce BIGRec [1], one of the most recent LRS, which serves as the backbone model in our method.

3.1 Evaluation of Item-side Fairness

Given a user with historical interactions, recommendation models usually rank all item candidates and return top-K items as recommendations [53]. To pursue fair recommendations across item groups, IF requires that the recommendation proportion of an item group G should be calibrated to the proportion of G in the users' historically liked items [27, 43]. In other words, recommendation models should neither over-recommend an item group nor decrease its recommendations compared to users' historical interactions. To ensure the protected group's rights and interests are not violated.

Formally, let $\mathcal{H}$ denote the set of all users' interaction sequences in the history, $\mathcal{P}$ denote the set of top-K recommendations of all users at the inference phase, and $\mathcal{G}$ denote the set of item groups. Given an item group $G \in \mathcal{G}$, we can measure the recommendation proportion of group G by

$$\mathbf{GP}(G) = \frac{\sum_{P\in\mathcal{P}}\sum_{v\in P}\mathbb{I}(v \in G)}{\sum_{G'\in\mathcal{G}}\sum_{P\in\mathcal{P}}\sum_{v\in P}\mathbb{I}(v \in G')}, \tag{1}$$

where $\mathbb{I}(v \in G)$ is an identity function:

$$\mathbb{I}(v \in G) = \begin{cases} 1, & \text{item } v \text{ belongs to group } G \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

Intuitively, $\mathbf{GP}(G)$ calculates the recommendation proportion of group G in the top-K recommendations of all users.

Accordingly, the interaction proportion of group G in the historical interaction sequences $\mathcal{H}$ can be obtained by:

$$\mathbf{GH}(G) = \frac{\sum_{H\in\mathcal{H}}\sum_{v\in H}\mathbb{I}(v \in G)}{\sum_{G'\in\mathcal{G}}\sum_{H\in\mathcal{H}}\sum_{v\in H}\mathbb{I}(v \in G')}. \tag{3}$$

Thereafter, we can define **Group Unfairness (GU)** *w.r.t.* group G to measure whether recommendation models amplify the recommendations of group G or overlook its recommendations. Formally,

$$\mathbf{GU}(G) = \mathbf{GP}(G) - \mathbf{GH}(G), \tag{4}$$

where $\mathbf{GU}(G) > 0$ suggests that the recommendation model tends to over-recommend items from group G, while $\mathbf{GU}(G) < 0$ implies the recommendation model overlooks group G. Both reflect item-side unfairness at the group level.

Following prior studies [27, 43], we adopt two evaluation metrics to aggregate the item-side unfairness across groups.

- **Mean Group Unfairness (MGU)** evaluates the overall fairness:

$$\mathbf{MGU} = \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}}|\mathbf{GU}(G)|, \tag{5}$$

- **Disparity Group Unfairness (DGU)** measures the disparity between the maximum and minimum GU across the groups in $\mathcal{G}$, which evaluates the upper bound of item-side unfairness:

$$\mathbf{DGU} = \max\{\mathbf{GU}(G)\}_{G\in\mathcal{G}} - \min\{\mathbf{GU}(G)\}_{G\in\mathcal{G}}. \tag{6}$$

In the experiments, we adopt the all-ranking protocol [53], where we can vary K to obtain top-K recommendations so that we have $\mathbf{GP}@K$, $\mathbf{GU}@K$, $\mathbf{MGU}@K$, and $\mathbf{DGU}@K$, respectively.

3.2 Brief on BIGRec

Extensive research on LRS is emerging. To ensure the recommendation quality of LRS, substantial work has demonstrated that instruction-tuning is an indispensable phase [1, 2]. Existing work often formulates recommendation data in natural language and tunes LLMs to generate personalized recommendations [1, 15, 56, 57]. In these studies, we select the representative BIGRec [1] to assess IF in LRS. BIGRec is selected for its simplicity yet high effectiveness, and, more importantly, BIGRec embodies the fundamental elements of instruction-tuning-based LRS and many methods can be formulated as further extensions based on the paradigm of BIGRec.

Following the general paradigm of applying LLMs in recommendation, BIGRec first represents user-item interaction data in natural language and employs instruction-tuning to fine-tune LLMs. In the inference stage, BIGRec generates item descriptions (*e.g.,* titles) as recommendations. However, the generated item descriptions might not exactly match any existing item. To deal with this issue, BIGRec designs a simple grounding paradigm, which leverages L2 embedding distance to ground generated item descriptions to existing items. Specifically, given **oracle** and $\mathbf{emb}_i$ denoting the token embeddings (*i.e.,* latent representations) of the generated descriptions and the description of item i, respectively, BIGRec calculates their L2 distance by:

$$D_i = ||\mathbf{emb}_i - \mathbf{oracle}||_2. \tag{7}$$

Thereafter, BIGRec ranks items based on this L2 distance and returns the K-nearest items as the recommendations.

4 PROBE THE ITEM-SIDE FAIRNESS OF LRS

To investigate the issue of item-side fairness in LRS, we initiated a preliminary experiment in this section by addressing the following research questions.

- **RQ1:** How does LRS perform in terms of item-side fairness compared to traditional recommendation models?
- **RQ2:** What is the root cause of the fairness issue in the LRS?

4.1 Experiment Setting

4.1.1 Datasets. We conducted experiments on two datasets containing information about different genres of items to perform an analysis of fairness regarding these genres.

- **MovieLens1M** [13][1] is a widely used dataset for a movie recommendation, which holds users' interaction records and movie genre. We retain titles as the textual information about films.
- **Steam** [20][2] provides user reviews for video games on the Steam platform. We adopt game titles as their text descriptions.

For both datasets, to better simulate real-world sequential recommendation scenarios, and prevent data leakage [9], we divide each dataset into 10 periods based on the timestamp of the interactions. Subsequently, we split the periods of datasets into training, validation, and testing sets using a ratio of 8:1:1. To save computing resources, we adopt the same approach as BIGRec, wherein we sample 65536 instances for training without altering the distribution of the original training dataset. Specifically, for the Steam dataset, we discovered a pronounced imbalance in the distribution of game genres in terms of user interactions. This imbalance results in the system primarily recommending games from these specific genres, which has a minimal impact on the overall metrics. Therefore, to minimize the Training Burden Without Altering the Conclusions, we made two adjustments. First, we removed games from the dataset whose respective genres had less than 10,000 interactions overall. This helped us focus on genres with substantial user engagement. Second, we set a maximum limit of 10 user interactions for both datasets to conduct our experiments and perform a

[1] https://grouplens.org/datasets/movielens/1m/.
[2] https://jmcauley.ucsd.edu/data/amazon/.

Table 1: Performance comparison of IF for groups split by popularity. The best results are indicated in boldface. We conduct each experiment three times and the average results are reported.

Dataset	Model	MGU@1	MGU@5	MGU@10	MGU@20	DGU@1	DGU@5	DGU@10	DGU@20
MovieLens1M	SASRec	**0.0238**	**0.0279**	0.0287	0.0279	**0.0846**	0.0959	0.0976	0.0936
	BIGRec	0.1044	0.0285	**0.0178**	**0.0128**	0.3917	**0.0956**	**0.0490**	**0.0415**
Steam	SASRec	**0.0173**	**0.0166**	**0.0162**	**0.0157**	**0.0647**	**0.0485**	**0.0411**	**0.0445**
	BIGRec	0.0411	0.0590	0.0736	0.0900	0.1586	0.1987	0.2460	0.3050

thorough analysis. The final statistical information of the dataset and other details are available in Appendix A.

To undertake a thorough investigation into the issue of item-side fairness in LRS, we have implemented two distinct categorizations for partitioning the items in our dataset. Consequently, we will analyze the extent of bias in the LRS using each of these partitioning schemes separately. This approach allows us to conduct a more comprehensive examination of the issue and gain deeper insights into the fairness dynamics within the system.

- **Popularity:**We analyzed the user interaction history by tallying the number of interactions for each item. Subsequently, we ranked the items based on their respective interaction counts. To ensure equitable distribution, we divided these items into five groups, with an equal number of items in each group.
- **Genre:** We extract the genre assigned to each item in the dataset and categorize them accordingly.

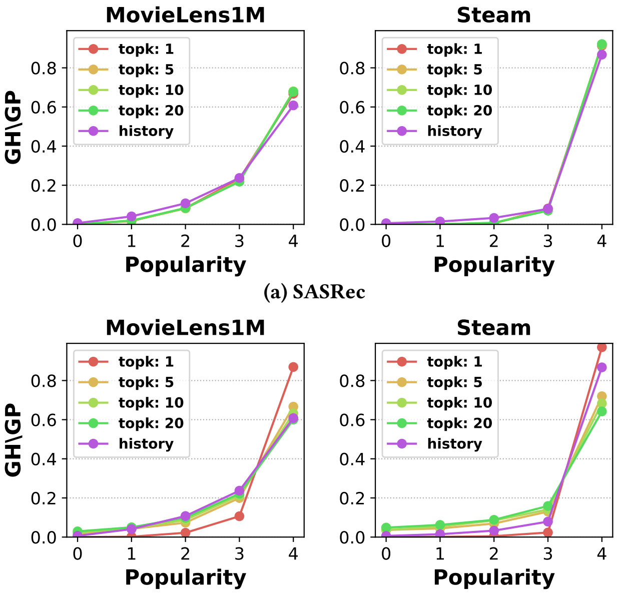
4.1.2 Compared Method. To thoroughly investigate IF in LRS, apart from our implementation of LRS in Section 3.2, we have developed and implemented a typical traditional recommendation system for comparison. In detail, we select **SASRec** which is commonly used in previous analyses of IF in the field of recommendation [17]. It employs a self-attention mechanism, specifically utilizing a left-to-right attention mechanism. This allows the model to acquire knowledge of sequential patterns and make accurate predictions about subsequent items.

4.2 Performance on Item-side Fairness

According to the aforementioned partitioning methods for items in the dataset, we have subdivided the **RQ2** into two distinct aspects: does LRS provide fairer recommendation results for different groups based on item popularity and genre compared to traditional recommendation systems?

4.2.1 Popularity Division. First, we conducted a comparative analysis of item popularity distributions between the recommendation results of two recommendation models and the item popularity distribution observed in the user interaction history on the test machine. Our focus was to highlight the discrepancies between these two distributions. The results are presented in Table 1. The main findings are as follows:

- Table 1 reveals that the fairness performance of the LRS-based model (BIGRec) is significantly inferior to that of the traditional SASRec model on the steam dataset and when considering the top recommendations for the MovieLens1M dataset, BIGRec also shows worse IF performance than SASRec.
- The recommendation model based on LLM demonstrates a higher degree of variation in fairness metrics compared to traditional models when considering different values of K. We attribute this phenomenon to the second step of grounding in BIGRec. In Section 4.3 of our research, we delve into this issue and conduct a comprehensive investigation to gain a deeper understanding of its implications.



(a) SASRec

(b) BIGRec

Figure 1: The proportion distribution of different groups divided by popularity in the top-K recommendation results, compared with the proportion distribution of different groups in the historical interactions (purple curve).

To gain a comprehensive understanding of the unfair issues faced by the popularity group within the LRS, we have analyzed two datasets. Figure 1 visually presents our findings. The results indicate that both MovieLens1M and Steam, the top-1 recommendation of LRS excessively recommended group with the highest level of popularity. And the proportions of groups 0 to 3 (representing unpopular items) are consistently lower compared to their historical sequences. When considering larger values of K, we can observe that this phenomenon is significantly alleviated in both datasets. We attribute this to the efficacy of grounding, and will further discuss this in the subsequent Section 4.3.

4.2.2 Genre Division. Subsequently, we evaluate the fairness of SASRec and BIGRec for different genre groups, and the comparison results are reported in Table 2. The main observations are as follows:

- The SASRec model achieves superior performance compared to the BIGRec model on almost all metrics. This indicates that, in comparison to conventional models, LRS exhibits significant disparities in fairness within the item genre grouping.
- In contrast to the inequities observed in popularity, the disparities in item genres are significantly less pronounced. This could be

Table 2: Performance comparison of IF for groups split by genre. The best results are indicated in boldface. We conduct each experiment three times and the average results are reported.

Dataset	Model	MGU@1	MGU@5	MGU@10	MGU@20	DGU@1	DGU@5	DGU@10	DGU@20
MovieLens1M	SASRec	**0.0048**	**0.0037**	**0.0031**	**0.0025**	**0.0224**	**0.0194**	**0.0172**	**0.0152**
	BIGRec	0.0068	0.0072	0.0065	0.0060	0.0374	0.0418	0.0382	0.0383
Steam	SASRec	**0.0122**	**0.0105**	0.0092	**0.0075**	**0.0477**	**0.0452**	**0.0442**	0.0403
	BIGRec	0.0158	0.0106	**0.0082**	0.0081	0.0487	0.0496	0.0443	**0.0341**

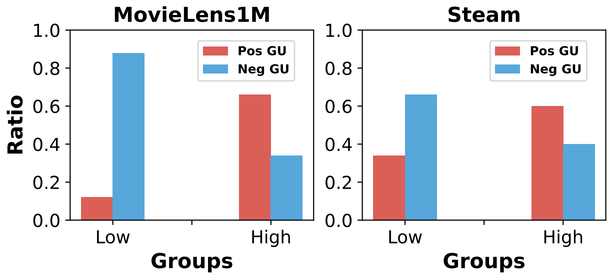
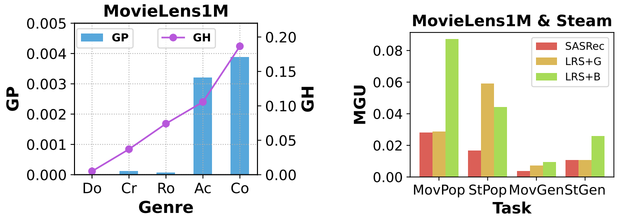


Figure 2: Comparison of GU@1 between groups divided by genre. We split these genre groups into two parts based on their interaction proportions in historical sequences, and each part has the same number of groups. "Pos GU" denotes GU@1 > 0, and "Neg GU" denotes GU@1 < 0. We can observe that high-popularity genres would be over-recommended (Pos GU), while low-popularity genres tend to be overlooked (Neg GU).



(a) Proportions of different groups *w.r.t.* genre in top-1 recommendation result, after deleting items of the corresponding genres at training phase, on MovieLens1M data.

(b) Unfairness (MGU@5) comparison among SASRec, LRS with grounding for ranking (LRS + G), and LRS with beam-serach for ranking (LRS + B).

Figure 3: Notations of (a): *GH* and *GP* denotes the proportions of groups in historical interactions and recommendation results respectively. The horizontal axis represents movie genres, where "Do", "Cr", "Ro", "Ac", and "Co" denote *Documentary*, *Crime*, *Romance*, *Action*, and *Comedy* respectively, whose *GH*s increase from left to right. Notations of (b): the horizontal axis represents different tasks, where "Mov" and "St" denote MovieLen1M and Steam datasets respectively; "Pop" and "Gen" denote groups divided by popularity and genre respectively.

attributed to the larger variety of item genres and their inherently smaller biases.

4.3 Cause of the Fairness Issues in LRS

In order to enhance our comprehension of the reason why LRS is unfair, we have devised experimental demonstrations in this section to discover its root reason. Firstly, we focus our attention on the performance of LRS in terms of top-1 recommendation result, which is slightly affected by grounding and shows the fairness issues inherently existing in LLMs themselves [3]. According to Figure 2, we can observe that LRS tends to overly focus on groups with high popularity and excessively recommends items from these groups. When considering the item genre, we organized them into groups based on the number of historical interactions. Subsequently, we evenly divided these groups into two parts: "Low" and "High," ensuring that each part contained an equal number of item genres. The result depicts that similarly to grouping by popularity, we can observe the LRS's tendency to disproportionately recommend item types that account for a larger proportion of historical interactions. Besides, we find that different from the analysis of the popularity group, there are groups with low proportions of group historical interactions that have positive group unfairness. This means that for genre groups with low frequencies, the model suggests a greater number of items compared to their historical trends.

To further explore why that phenomenon occurs across various genre groups, we propose additional experiments to uncover these factors. Specifically, we select five genres in MovieLens1M and remove them from our training set. Then, we verify the fairness metrics of our newly trained LLM on these genre groups as illustrated in Figure 3a. Surprisingly, we found that even after certain item types were removed, there was still a small probability that they would be recommended. For instance, even if during the supervised fine-tuning stage, models had not encountered movies of the Comedy genre, they could still recommend "Mighty Ducks, The (1992)" to users[4]. This indicates that during the recommendation process, the models leverage knowledge acquired from their pre-training phase, which potentially affects the fairness of their recommendations.

Then, we shall explore what happens as the number of recommended items increases. The primary experimental observations and conclusions are as follows:

- As shown in Figure 1, there is a clear decrease in the proportion of group 4 with the increase of the value of top-K, while simultaneously observing an increase in the proportions of groups 0 to 3. Thus we can conclude that the inequity in the popularity of the model decreases as K increases. We attribute this phenomenon to the fact that the original grounding step of BIGRec is not affected by the influence of popularity in specific datasets and consequently recommends a plethora of unpopular items. However, this also results in the prejudice of the model towards low-popularity groups transfer to high-popularity groups.

[3] The reason and details can be found in Appendix B.

[4] More cases and details can be found in Appendix C.

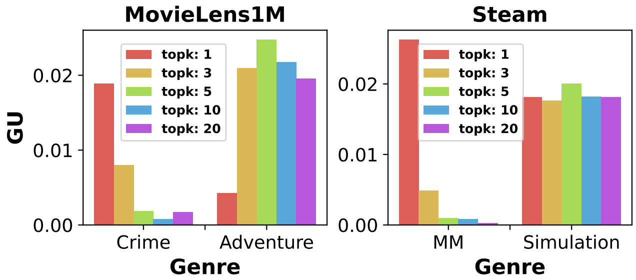


Figure 4: The GU (Group Unfairness) of different groups divided by genres in top-K recommendation results.

- Besides, according to Figure 4, we find that due to the non-uniformity of the embedding space, this step serves to mitigate the inherent unfairness present in certain genres. However, it is important to note that this process may introduce new fairness concerns for other genres or maintain the status. This indicates that the impact of the direct grounding on fairness across different genre groups will vary depending on the distribution of these groups in the embedding space.
- Based on our findings, we are considering whether it would be advisable to eliminate the second step of BIGRec and rely solely on beam search to address the issue discussed. To gain a deeper understanding of this matter, we conducted an analysis, as depicted in Figure 3b[5]. The results of our experiments indicate that the use of beam search significantly amplifies inequity, except for the popularity bias observed in the Steam dataset, which can be attributed to the over-correction of BIGRec. This phenomenon is not unique, as it shares similarities with previous discussions in the field of Natural Language Processing. Employing beam search exacerbates various biases inherent in the model [37, 39].

4.4 Summary

Based on the experiments and phenomena mentioned above, we can deduce the following conclusions:

- No matter whether observing the grouping of popular items or the grouping of item genres, we have observed unfairness on the item side in LRS. This unfair phenomenon is particularly evident in the top recommendation.
- When contemplating the top 1 recommendation, it becomes evident that LRS has a propensity to excessively promote categories with a higher proportion in the training set, and this results in a diminished allocation of recommendations towards items that were originally overlooked.
- LRS can recommend item genres that are never seen in the supervised instruction tuning period, which means the IF in LRS is not only derived from the training set during the fine-tuning phase but also emanates from the semantic priors obtained during the pre-training phase.

5 METHODS

To enhance the IF of LRS, we propose the IFairLRS framework. As discussed in Section 4.3, the unfairness stems from two phases: the instruction tuning phase of LLMs (*i.e.,* in-learning stage) and the ranking phase of the recommendation system (*i.e.,* post-learning stage). Accordingly, IFairLRS has two specific strategies: reweighting and reranking strategies to enhance the IF of LRS during the in-learning and post-learning stages, respectively.

5.1 Reweighting Strategy

We first consider reweighting the instruction-tuning samples to reduce the effect of biased training data. Traditional IF methods might add regulation terms [16, 55] or adaptively adjust the sample weights [40] during the in-learning stages of conventional recommendation models. However, these methods require the ranking scores over the items of different groups [16, 55] or adaptive calculate the weight of various groups [40]. It is challenging for generative LRS to fulfill these requirements since generative LRS is tuned to maximize the likelihood of the tokens of target item descriptions, instead of calculating the ranking scores over many item candidates like discriminative models.

In light of this, we pre-calculate the sample weights for reweighting before instruction-tuning. Specifically, for the instruction-tuning, we split users' historical interaction sequences $\mathcal{H}$ into the set of training sequences $\mathcal{H}_{tr}$ and the set of target items $\mathcal{H}_{ta}$, where the last item of each sequence is treated as the target item and (H_i, T_i) with $H_i \in \mathcal{H}_{tr}$ and $T_i \in \mathcal{H}_{ta}$ represents the i-th instruction-tuning sample. If more target items belong to an item group G in $\mathcal{H}_{ta}$, LRS might be tuned biased to this group. As shown in 4.2.1, the LLM could learn more bias stemming from unbalanced groups. As such, we adjust the sample weights of different groups based on their proportions in $\mathcal{H}_{tr}$ and in $\mathcal{H}_{ta}$.

In detail, we can calculate the interaction proportion $\mathbf{GH}(G)$ of a group G in $\mathcal{H}_{tr}$ and $\mathcal{H}_{ta}$ by Equation (3), which are denoted as $\mathbf{GH}_{tr}(G)$ and $\mathbf{GH}_{ta}(G)$, respectively. Given $\mathbf{GH}_{tr}(G)$ and $\mathbf{GH}_{ta}(G)$, we calculate the weight W_G of group G by:

$$W_G = \frac{\mathbf{GH}_{tr}(G)}{\mathbf{GH}_{ta}(G)}, \tag{8}$$

where $W_G < 1$ suggests that there is an amplified proportion of group G in $\mathcal{H}_{ta}$ than that in $\mathcal{H}_{tr}$, and thus we need to reduce the weight of group G in the instruction-tuning stage. Moreover, since an item may belong to several groups, the weight W_i of the i-th sample (H_i, T_i) can be calculated as:

$$W_i = \frac{1}{|\mathcal{G}_i|} \sum_{G \in \mathcal{G}_i} W_G, \tag{9}$$

where $\mathcal{G}_i$ denotes the groups that the target item T_i belongs to. Finally, we utilize W_i to reweight each sample and the loss of the i-th sample (H_i, T_i) is as follows:

$$\mathbf{L}(H_i, T_i) = W_i * \mathcal{L}(H_i, T_i), \tag{10}$$

where $\mathcal{L}(\cdot, \cdot)$ denotes the instruction-tuning loss of LLMs.

5.2 Reranking Strategy

In addition to the reweighting strategy for the in-learning stage, we consider reranking the items in the post-learning stage. Given the top-K recommendations from LRS, the reranking strategy can calculate $\mathbf{GU}(G)@K$ to measure the unfairness of each group G. To enhance IF, we can rerank the items of different groups by revising $\mathbf{GU}(G)@K$ as a punishment term.

[5] Due to the significant cost associated with beam inference, we have specifically compared the circumstances at iteration 5 only.

Table 3: The IF of IFairLRS with different strategies for groups split by popularity. The best results of all methods are indicated in boldface. IFairLRS(RW) denotes the reweighting strategy and IFairLRS(RR) denotes the reranking strategy. We conduct each experiment three times and report the average results.

Dataset	Model	Fairness						Accuracy	
		MGU@1 ↓	**MGU@5** ↓	**MGU@20** ↓	**DGU@1** ↓	**DGU@5** ↓	**DGU@20** ↓	**NDCG@5** ↑	**HR@5** ↑
MovieLens1M	BIGRec	0.1044	0.0285	0.0128	0.3917	0.0956	0.0415	**0.0299**	0.0345
	IFairLRS(RW)	**0.0990**	**0.0198**	0.0160	**0.3585**	**0.0527**	0.0544	0.0295	0.0341
	IFairLRS(RR)	0.1074	0.0352	**0.0125**	0.4011	0.1190	**0.0413**	0.0297	**0.0348**
Steam	BIGRec	0.0411	0.0590	0.0900	0.1586	0.1987	0.3050	0.0311	0.0346
	IFairLRS(RW)	**0.0371**	0.0691	0.0959	**0.1410**	0.2290	0.3208	0.0311	0.0341
	IFairLRS(RR)	0.0417	**0.0179**	**0.0377**	0.1610	**0.0774**	**0.1308**	**0.0312**	**0.0348**

Specifically, we calculate $\mathbf{GU}(G)@K$ of each group G *w.r.t.* varying K values and then aggregate them to obtain the punishment term $\mathbf{U}_G$ for group G. We vary K to calculate $\mathbf{U}_G$ because the grounding strategy of LRS might over-correct the bias when K is large (as shown in Section 4.3). By aggregating $\mathbf{GU}(G)@K$ for punishment, we can better regulate the unfairness across different top-K recommendations. Formally, we have

$$\mathbf{U}_G = \sum_{K\in\mathcal{K}} \gamma_K * \mathbf{GU}(G)@K, \tag{11}$$

where $\mathcal{K}$ includes the possible values of K and γ_K adjusts the weights of unfairness with different top-K values. In this work, we use $\gamma_K = K/\sum_{K'\in\mathcal{K}} K^{'}$, where γ_K increases with K increasing. In this way, it pays more attention to alleviating the over-correction of the grounding strategy of LRS when K is large. Notably, we further normalize the $\mathbf{U}(G)$ into $[-1, 1]$ by

$$\hat{\mathbf{U}}_G = \frac{\mathbf{U}_G}{\max\{|\mathbf{U}_{G'}|\}_{G'\in\mathcal{G}}}, \tag{12}$$

where $|\mathbf{U}_{G'}|$ takes the absolute value of $\mathbf{U}_{G'}$. Since an item may belong to several groups, we calculate the punishment $\hat{\mathbf{U}}_i$ of the i-th item by:

$$\hat{\mathbf{U}}_i = \frac{1}{|\mathcal{G}_i|}\sum_{G\in\mathcal{G}_i} \hat{\mathbf{U}}_G, \tag{13}$$

where $\mathcal{G}_i$ denote the groups that the i-th item belong to. Eventually, we add $\hat{\mathbf{U}}_i$ to Equation (7) for item reranking:

$$\tilde{D}_i = \frac{D_i}{(1-\hat{\mathbf{U}}_i)^{\alpha}}, \tag{14}$$

where a hyperparameter $\alpha > 0$ is to regulate the influence of punishment, and D_i denotes the L2 distance between the embedding of the i-th item and the embedding of the generated item description by LLMs. Intuitively, the items with positive $\hat{\mathbf{U}}_i$ are over-recommended (see explanation in Equation 4) and thus they get a larger L2 distance $\tilde{D}_i$ in Equation 14 for lower recommendation probabilities. In contrast, the items with negative $\hat{\mathbf{U}}_i$ will have higher recommendation probabilities with smaller $\tilde{D}_i$.

6 EXPERIMENT

In this section, we conduct experiments on the two datasets to answer the following research question:

- **RQ3:** Can IFairLRS with two strategies effectively enhance the IF of LRS?

6.1 Experiment Setting

Following Section 4.1.1, we conduct experiments on two datasets: MovieLens1M and Steam. The dataset statistics and division can be found in Section 4.1.1. In this section, we not only evaluate the fairness but also evaluate the accuracy of recommendation models, so as to examine whether IFairLRS will damage the recommendation accuracy of LRS while pursuing IF. Following the previous work [1, 53], we use two commonly used evaluation metrics for accuracy measurements: Normalized Discounted Cumulative Gain (**NDCG**) and Hit Ratio (**HR**). Both metrics are computed under the all-ranking protocol [53].

We implement all of the methods using PyTorch. We follow the setting of BIGRec using LLaMA [44] as the base LLM for instruction-tuning. We use the Adam [18] as the optimizer with a learning rate of $1e-3$, a batch size of 128, and we tune the weight decay in the range of $[1e-2, 1e-3, 1e-4, 1e-5, 1e-6]$. For the reranking strategy of IFairLRS, we tune its hyperparameter α in Equation (14) in the range of [0,0.1] with step 0.01. We utilize the validation sets to calculate **GU**(G) for Equation 11 because it includes user interactions temporally closer to the testing sets.

6.2 Performance Comparison

To answer **RQ3**, we compare the fairness and accuracy of BIGRec with the reweighting and reranking strategies of IFairLRS. Specifically, we divide item groups in two ways: popularity and genre, and then measure the recommendation fairness and accuracy in terms of **MGU@**K, **DGU@**K, **NDCG@**K, and **HR@**K.

6.2.1 *Popularity Division.*

To validate the effectiveness of IFairLRS on diverse popularity groups, we compare BIGRec with the reweighting strategy and reranking strategy of IFairLRS. The fairness and accuracy comparison is reported in Table 3, from which the main observations are as follows:

- The reweighting strategy in IFairLRS proves effective in enhancing the IF of LRS. On the MovieLens1M dataset, this strategy yields the best **MGU@**1 and **DGU@**1, showcasing improvements of 5% and 8.4% over BIGRec, respectively. Similarly, on the Steam dataset, the reweighting strategy demonstrates fairness improvements of 9.7% and 11.1% on **MGU@**1 and **DGU@**1, respectively. These findings indicate that the reweighting strategy successfully promotes the calibration between top-1 recommendations and users' historical interactions across item groups.
- The reranking strategy reveals better fairness when K is large while its effectiveness is limited with small K. For instance,

Table 4: The IF of IFairLRS with different strategies for groups split by genre. The best results of all methods are indicated in boldface. IFairLRS(RW) denotes the reweighting strategy and IFairLRS(RR) represents the reranking strategy. We conduct each experiment three times and report the average results.

Dataset	Model	Fairness						Accuracy	
		MGU@1 ↓	**MGU**@5 ↓	**MGU**@20 ↓	**DGU**@1 ↓	**DGU**@5 ↓	**DGU**@20 ↓	**NDCG**@5 ↑	**HR**@5 ↑
MovieLens1M	BIGRec	0.0068	0.0072	0.0060	0.0374	0.0418	0.0383	**0.0299**	**0.0345**
	IFairLRS(RW)	**0.0050**	0.0067	0.0056	**0.0268**	0.0359	0.0366	0.0291	0.0337
	IFairLRS(RR)	0.0089	**0.0054**	**0.0036**	0.0439	**0.0339**	**0.0184**	0.0291	0.0336
Steam	BIGRec	0.0158	0.0106	0.0081	0.0487	0.0496	0.0341	0.0311	0.0346
	IFairLRS(RW)	**0.0138**	0.0092	0.0078	**0.0483**	0.0440	0.0331	**0.0316**	**0.0348**
	IFairLRS(RR)	0.0184	**0.0084**	**0.0065**	0.0541	**0.0396**	**0.0284**	0.0311	0.0347

IFairLRS(RR) has comparable or slightly worse fairness *w.r.t.* **MGU**@1 and **DGU**@1 on two datasets while it shows better performance *w.r.t.* **MGU**@20 and **DGU**@20, especially on the Steam dataset. The possible reason is that reranking can better balance the fairness across groups when K is large with more recommendation slots.

- BIGRec and IFairLRS with two strategies demonstrate comparable accuracy in terms of **NDCG**@5 and **HR**@5. The performance difference among BIGRec, IFairLRS(RW), and IFairLRS(RR) for **NDCG**@5 on both datasets fluctuates within 1%. Similarly, for **HR**@5, the deviation remains within 5%. These results validate that IFairLRS(RW) and IFairLRS(RR) do not enhance IF at the significant expense of recommendation accuracy.
- As the value of top-K increases, the observed changes in metrics for IFairLRS(RW) remain consistent with the trend of BIGRec. Even on the **MGU**@20 and **DGU**@20, the IFairLRS with the reweighting strategy obtains worse fairness for various popularity groups. Hence, the issues that exist in the grounding are not solved by the reweighting strategy.
- The IFairLRS(RR) obtains comparable or slightly worse **MGU**@1 and **DGU**@1. We analyze the reasons for this phenomenon. For instance, as shown in Figure 1, the **GU**@K of group 4 reduces from positive to negative with the increase of the value of top-K on the Steam dataset. We vary K to calculate the punishment of the group and the **GU**@K with higher K has more weight. Hence, the punishment of group 4 is the opposite of GU@1, which leads to the IFairLRS(RR) obtaining comparable or slightly worse **MGU**@1 and **DGU**@1.

6.2.2 Genre Division. To validate the effectiveness of the reweighting strategy and the reranking strategy of IFairLRS on different genre groups, we conduct experiments to compare their performance. The fairness and accuracy performance is reported in Table 4. In the table, we have the observations as follows:

- Both strategies exhibit the effectiveness of improving the fairness of LRS. On MovieLens1M, the reweighting strategy enhances 0.0018 for **MGU**@1 and 0.0106 for DGU@1 over BIGRec (as much as 30% and 28.3%, respectively). On Steam, IFairLRS with the reweighting strategy obtains 12.7% and 11.1% improvement on **MGU**@1 and **DGU**@1, respectively. Furthermore, on MovieLens1M, IFairLRS with the reranking strategy obtains the best **MGU**@20 and **DGU**@20, which outperform the BIGRec 40% and 52%, respectively. The **MGU**@20 and the **DGU**@20 of IFairLRS with the reranking strategy are 0.0016 and 0.0057 lower than the BIGRec (as much as 19.7% and 16.7%) on Steam. These suggest that the reranking strategy could also mitigate the unfairness across various genre groups when K is large.
- Both strategies improve the fairness of LRS without compromising the recommendation accuracy. On MovieLens1M, two strategies have the accuracy decrease by up to 0.001 *w.r.t.* **NDCG**@5 and 0.0012 *w.r.t.* **HR**@5. On Steam, **NDCG**@5 and **HR**@5 fluctuate within 3%. This validates that although IFairLRS adjusts the recommendation proportion of different item groups for fairness, it does not reduce the number of recommended positive items (*i.e.,* users' liked items).

7 CONCLUSION

In this work, we examined the item-side unfairness of LRS and compared it with that of conventional recommendation models. We found that LRS is not only significantly influenced by the popularity factor but also affected by the inherent semantic biases within LLMs. These findings highlight the necessity of improving the IF of LRS. To achieve this goal, we conducted a preliminary exploration by proposing a concise and effective framework called IFairLRS. IFairLRS adapts the reweighting and reranking strategies from traditional IF methods to the in-learning and post-learning stages of LRS, respectively. We conducted extensive experiments on two real-world datasets, MovieLens1M and Steam, demonstrating the effectiveness of our proposed IFairLRS in improving the IF of LRS without sacrificing the recommendation accuracy.

We firmly believe that enhancing the IF of LRS is vitally important for improving the trustworthiness of LRS and contributing to making the Web for good. IFairLRS with two strategies presents a preliminary exploration, and it is promising to design more effective fairness-oriented methods specifically tailored for LRS in future work. Moreover, our analysis has found that the pre-training of LLMs affects the IF of LRS while it is challenging to quantify and alleviate such impact, leaving ample room for future research. Lastly, we mainly investigate the IF of LRS based on the group division of popularity and genre. Future efforts might shed light on the fairness of more group divisions (*e.g.,* different item uploader groups) and LLMs, individual-level fairness, and long-term fairness.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2022YFB3104701), the National Natural Science Foundation of China (U21B2026 and 62272437), and the CCCD Key Lab of Ministry of Culture and Tourism.

REFERENCES

[1] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Fuli Feng, Xiangnan He, and Qi Tian. 2023. A Bi-Step Grounding Paradigm for Large Language Models in Recommendation Systems. *CoRR* abs/2308.08434 (2023).

[2] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *RecSys*. ACM, 1007–1014.

[3] Keqin Bao, Jizhi Zhang, Yang Zhang, Wang Wenjie, Fuli Feng, and Xiangnan He. 2023. Large Language Models for Recommendation: Progresses and Future Directions. In *SIGIR-AP*.

[4] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*. 405–414.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *NeurIPS*. Curran Associates, Inc., 1877–1901.

[6] Robin Burke. 2017. Multisided Fairness for Recommendation. *CoRR* abs/1707.00093 (2017).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018).

[8] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender Systems in the Era of Large Language Models (LLMs). *CoRR* abs/2307.02046 (2023).

[9] Xinyan Fan, Jianxun Lian, Wayne Xin Zhao, Zheng Liu, Chaozhuo Li, and Xing Xie. 2022. Ada-Ranker: A Data Distribution Adaptive Ranking Paradigm for Sequential Recommendation. In *SIGIR*. 1599–1610.

[10] 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*. 445–453.

[11] Yingqiang Ge, Xiaoting Zhao, Lucia Yu, Saurabh Paul, Diane Hu, Chu-Cheng Hsieh, and Yongfeng Zhang. 2022. Toward Pareto efficient fairness-utility trade-off in recommendation through reinforcement learning. In *WSDM*. 316–324.

[12] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. 2010. Social Media Recommendation Based on People and Tags. In *SIGIR*. 194–201.

[13] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *TIIS* (2016), 19:1–19:19.

[14] T. Ryan Hoens, Marina Blanton, and Nitesh V. Chawla. 2010. Reliable medical recommendation systems with patient privacy. In *IHI*, Tiffany C. Veinot, Ümit V. Çatalyürek, Gang Luo, Henrique Andrade, and Neil R. Smalheiser (Eds.). ACM, 173–182.

[15] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian J. McAuley, and Wayne Xin Zhao. 2023. Large Language Models are Zero-Shot Rankers for Recommender Systems. *CoRR* abs/2305.08845 (2023).

[16] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2018. Recommendation Independence. In *FAT (Proceedings of Machine Learning Research, Vol. 81)*. PMLR, 187–201.

[17] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. IEEE Computer Society, 197–206.

[18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[19] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *NeurIPS*.

[20] Himabindu Lakkaraju, Julian J. McAuley, and Jure Leskovec. 2013. What's in a Name? Understanding the Interplay between Titles, Content, and Communities in Social Media. In *ICWSM*. The AAAI Press.

[21] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*. Association for Computational Linguistics, 7871–7880.

[22] Jie Li, Yongli Ren, and Ke Deng. 2022. FairGAN: GANs-based fairness-aware learning for recommendations with implicit feedback. In *WWW*. 297–307.

[23] Ruyu Li, Wenhao Deng, Yu Cheng, Zheng Yuan, Jiaqi Zhang, and Fajie Yuan. 2023. Exploring the Upper Limits of Text-Based Collaborative Filtering Using Large Language Models: Discoveries and Insights. *arXiv preprint arXiv:2305.11700* (2023).

[24] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2023. Fairness in Recommendation: Foundations, Methods, and Applications. *TIST* (2023), 1–48.

[25] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. *CoRR* abs/2306.05817 (2023).

[26] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. ReLLa: Retrieval-enhanced Large Language Models for Lifelong Sequential Behavior Comprehension in Recommendation. *CoRR* abs/2308.11131 (2023).

[27] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2019. Crank up the Volume: Preference Bias Amplification in Collaborative Recommendation. In *RecSys*. CEUR-WS.org.

[28] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A Multi-facet Paradigm to Bridge Large Language Model and Recommendation. *CoRR* abs/2310.06491 (2023).

[29] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. *CoRR* abs/2304.10149 (2023).

[30] Weiqing Min, Shuqiang Jiang, and Ramesh C. Jain. 2020. Food Recommendation: Framework, Existing Solutions, and Challenges. *TMM* 22, 10 (2020), 2659–2671.

[31] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*. 429–438.

[32] Vincenzo Moscato, Antonio Picariello, and Antonio Maria Rinaldi. 2010. A Combined Relevance Feedback Approach for User Recommendation in E-commerce Applications. In *ACHI*, Ray Jarvis and Cosmin Dini (Eds.). IEEE Computer Society, 209–214.

[33] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023).

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS* 35 (2022), 27730–27744.

[35] Ioannis K. Paparrizos, Berkant Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *RecSys*, Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius (Eds.). ACM, 325–328.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[37] Nicholas Roberts, Davis Liang, Graham Neubig, and Zachary C. Lipton. 2020. Decoding and Diversity in Machine Translation. *CoRR* abs/2011.13477 (2020).

[38] Fatemeh Sarvi, Maria Heuss, Mohammad Aliannejadi, Sebastian Schelter, and Maarten de Rijke. 2022. Understanding and mitigating the effect of outliers in fair ranking. In *WSDM*. 861–869.

[39] Danielle Saunders, Rosie Sallis, and Bill Byrne. 2022. First the Worst: Finding Better Gender Translations During Beam Search. In *Findings of ACL*. 3814–3823.

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

[41] Qijie Shen, Wanjie Tao, Jing Zhang, Hong Wen, Zulong Chen, and Quan Lu. 2021. Sar-net: a scenario-aware ranking network for personalized fair recommendation in hundreds of travel scenarios. In *CIKM*. 4094–4103.

[42] Tianhao Shi, Yang Zhang, Zhijian Xu, Chong Chen, Fuli Feng, Xiangnan He, and Qi Tian. 2023. Preliminary Study on Incremental Learning for Large Language Model-based Recommender Systems. *CoRR* abs/2312.15599 (2023).

[43] Harald Steck. 2018. Calibrated Recommendations. In *RecSys*. Association for Computing Machinery, 154–162.

[44] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023).

[45] Jason Turcotte, Chance York, Jacob Irving, Rosanne M. Scholl, and Raymond J. Pingree. 2015. News Recommendations from Social Media Opinion Leaders: Effects on Media Trust and Information Seeking. *J COMPUT-MEDIAT COMM* (2015), 520–535.

[46] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative Recommendation: Towards Next-generation Recommender Paradigm. *CoRR* abs/2304.03516 (2023).

[47] Xuezhi Wang, Nithum Thain, Anu Sinha, Flavien Prost, Ed H Chi, Jilin Chen, and Alex Beutel. 2021. Practical compositional fairness: Understanding fairness in multi-component recommender systems. In *WSDM*. 436–444.

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

[49] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *TMLR* 2022 (2022).

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*.

[51] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A Survey on Large Language Models for Recommendation. *CoRR* abs/2305.19860 (2023).

[52] Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. *CoRR* abs/2306.10933 (2023).
[53] Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. 2023. A Generic Learning Framework for Sequential Recommendation with Distribution Shifts. In *SIGIR*. ACM, 331–340.
[54] Zhengyi Yang, Jiancan Wu, Yanchen Luo, Jizhi Zhang, Yancheng Yuan, An Zhang, Xiang Wang, and Xiangnan He. 2023. Large Language Model Can Interpret Latent Space of Sequential Recommender. arXiv:2310.20487 [cs.IR]
[55] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *NeurIPS*. 2921–2930.
[56] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation. In *RecSys*. ACM, 993–999.
[57] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as Instruction Following: A Large Language Model Empowered Recommendation Approach. *CoRR* (2023).
[58] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *SIGIR*. 11–20.
[59] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. CoLLM: Integrating Collaborative Embeddings into Large Language Models for Recommendation. *CoRR* abs/2310.19488 (2023).
[60] Wayne Xin Zhao, Kun Zhou, Junyi Li, et al. 2023. A Survey of Large Language Models. *CoRR* abs/2303.18223 (2023).
[61] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. 2010. The impact of YouTube recommendation system on video views. In *SIGCOMM*, Mark Allman (Ed.). ACM, 404–410.
[62] Ziwei Zhu, Jingu Kim, Trung Nguyen, Aish Fenton, and James Caverlee. 2021. Fairness among new items in cold start recommender systems. In *SIGIR*. 767–776.

A DATASETS DETAILS

Table 5: Statistics of the experimental datasets.

Datasets	#Items	#Interactions	#Sequences
MovieLens1M	3,883	1,000,209	939,809
Steam	14,662	7,793,036	1,620,946

The statistics of the experimental datasets can be seen in Table 5.

B RELATIONSHIP BETWEEN OUTPUTS OF LLM AND TOP 1 RECOMMENDATIONS

Table 6: The number of outputs of LLM existing in the datasets.

Existance	MovieLens1M	Steam
Yes	93667	160696
No	314	1399

As shown in Table 6, we find that most of the outputs generated by the LLM exist in the datasets. Because we use the $L2$ distance to ground the outputs from the recommendation space to the actual item space. So, the grounding step does not modify these outputs existing in the dataset in the top-1 recommendation system. The impact of grounding for the top-1 recommendation results is very slight. Hence we could utilize the results of the top-1 recommendations to represent the performance of the LLM.

C ANALYSIS OF DELETING GENRE ON MOVIELENS1M

We delete the items belonging to "Comedy" both in the historical interactions and target items in the training set. We find that the LRS continues to recommend movies in "Comedy". The outputs of LLM are two folds:

- The output of LLM is in the dataset. For example, the LLM generates the movie "Airplane! (1980)", which is a comedy movie and it is in the dataset.
- The output of LLM is not in the dataset. For instance, the LLM generates the movie "Mighty Ducks, The (1992)", which is not in the dataset, and the nearest movie in the embedding space is "The Mighty Ducks (1992)".

D ANALYSIS OF DIFFERENCE BETWEEN MOVIELENS1M AND STEAM

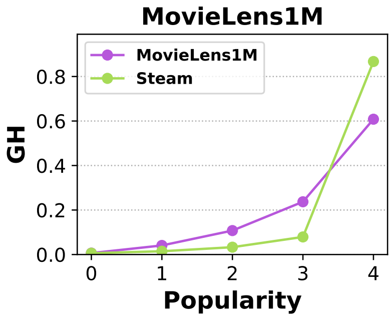


Figure 5: The proportions of various popularity groups of MovieLens1M and Steam.

As shown in Figure 5, we can observe that Steam has a larger popularity bias compared to MovieLens1M. The proportion of the most popular group in Steam is significantly higher than in MovieLens1M.