LEVERAGING HOLISTIC EXPLANATIONS TO MITIGATE POPULARITY BIAS FOR RECOMMENDER SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recommender systems often suffer from popularity bias, where items with high historical engagement ensure a dominant presence in the recommendation lists while equally relevant but less popular items (called niche items) remain under exposed towards majority of the users, thus impacting their reach within mainstream platforms. This bias arises partly due to the learning strategy of existing recommender models which display heavy reliance on interaction frequency and shallow contextual features that characterize any item, which fail to capture the true preferences of any users. To address this, we propose Expl-Debias, a novel framework that leverages holistic explanations to enrich user–item preference modeling and mitigate popularity bias. Expl-Debias operates in two stages: (Stage-1) a base training phase that learns general user–item utility, and (Stage-2) a contrastive explanation-aware training phase that incorporates LLM-generated positive and negative explanations to explicitly guide relevance learning toward personally aligned items and away from popular yet irrelevant ones. Extensive experiments on three real-world datasets demonstrate that our approach significantly improves recommendation accuracy while substantially reducing popularity bias, outperforming state-of-the-art LLM recommendation and debiasing baselines. These results demonstrate that integrating contrastive explanations offers an effective new direction for mitigating popularity bias in recommendation by balancing the tradeoff occurring between the recommendation performance and the negative effect of popularity bias. We provide our code at `https://anonymous.4open.science/r/Expl-Pop-Bias-089A/`.

1 INTRODUCTION

Over the recent years, recommender systems have been very relevant for practical uses to bridge the gap between users and products/services. Recommendation systems significantly enhance user experience by providing personalized suggestions justifying why the suggestions predicted by any recommender are presented to any user. Explanations offer advantages to all the different entities of a recommender such as the users, manufacturers/provisioners and developers (Zhang and Chen, 2020; Chen et al., 2022b). Manufacturers/Provisioners can use explanations to their advantage for increasing the visibility of their products/services by highlighting the important aspects which will attract the users for further interaction. Beyond user trust, explanations also play a critical role in evaluating and diagnosing the fairness of recommendation outcomes, highlighting potential biases or disparities within the system. In practice, system developers leverage explanations to a major extent in detecting/resolving any hidden issues existing within the model outcomes: such as bias disparities (Pan et al., 2021; Ge et al., 2022a), privacy leakage (Ghazimatin et al., 2020; Zhao et al., 2022) or model malfunctioning due to attacks (Fan et al., 2023; Tao et al., 2018) . From a practical perspective, explanations could be utilized for practical diagnosis of model outcome disparities.

However, there has been significant work raised in literature which validates the fact that most of the recommender algorithms exhibit algorithmic bias in their outcomes. In general, item-side biases are more implicit and they are much more difficult to detect. Additionally, item-side biases such as popularity biases present numerous critical consequences which can impact both the users and the manufacturers simultaneously (Zhao et al., 2025; Chen et al., 2021). Popularity bias occurs when recommendation systems disproportionately promote items already enjoying high engagement, thereby sidelining equally relevant but less popular alternatives, leading to a phenomenon

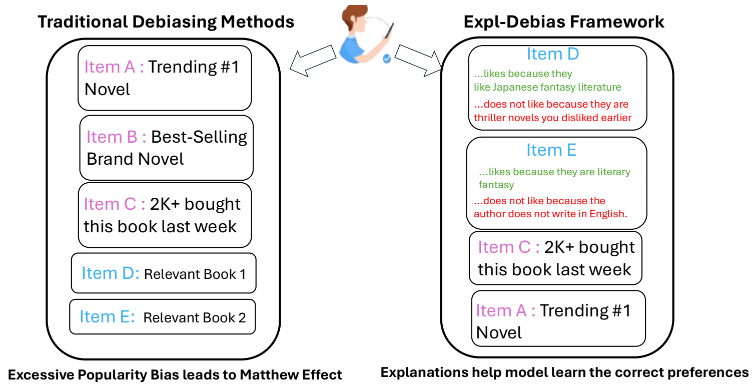


Figure 1: Comparison between the recommendation suggestions provided by traditional debiasing methods and our proposed Expl-Debias framework.

called Mathew's Effect (Abdollahpouri, 2019). This phenomenon significantly impacts user experience by limiting exposure to diverse content and products and by repeatedly reinforcing the most popular items within the existing market, thereby diminishing the user satisfaction. This effect could disadvantage smaller creators or businesses who could eventually be phased out and therefore reduce their involvement with the platform. Meanwhile, it bears a major influence in disappointing the providers since their products/services will also receive a limited reach across consumers which would negatively impact their viability. Therefore, there is a dire requirement to ensure fairness amongst the recommendation outcomes within any recommender algorithm.

There have been many methods which have inspected and mitigated item-side unfairness amongst recommender systems. In general, most of these methods have diagnosed the root cause attributed within recommendation algorithms being overly reliant on historical interaction data, inherently favoring items with higher visibility and prior engagement. The main reason behind such reliance is because traditional recommendation algorithms often lack a comprehensive modeling of user preferences that considers the true relevance of items (Deldjoo et al., 2022; Yalcin and Bilge, 2021). Although there have been previous works that have combined different forms of data which provide additional information about items, there has been paucity of methods which have considered the deeper relationship between any specific user and item.

In order to resolve this missing aspect, explanations can be leveraged to ensure clear input signals that specify why certain items are recommended or ignored. This is because explanations have the capabilities to specify both the pros and cons of each item towards any user in detail, thus providing a solid base for improving the recommendation quality. For example, consider a user browsing an online book database who frequently receives suggestions for most-popular/in-demand novels. A traditional recommender, relying merely on past purchase data, some contextual information and item popularity, may continue to promote these bestsellers, even if the user has recently shown interest in underexplored genres such as "translated Japanese fantasy literature." The quality of recommendations can be enhanced by leveraging explanations which would include direct display of the user's preferences towards such underexplored set of items. This behavior can be accentuated through explanation texts such as: "This book is suggested because it matches your recent queries about Japanese authors and literary fantasy fiction, while also slightly aligned with one of your disliked thriller novels." These explanations adds transparency within the model and it would help for easier recognition of personally relevant recommendations and mitigates the dominance of already-popular items in their recommendation list, as shown in Fig. 1. Therefore, using explanations while modeling the personalized preferences can facilitate the strong likes and dislikes of any user.

In this work, we seek to mitigate popularity bias existing within recommendation algorithms by enhancing the comprehensiveness of personalized preferences of each user. We focus our work primarily towards including explanations that describe completely about an item's pros and cons which would be introduced while modeling user-item interaction data. Our study addresses the lack of learning complete true user preferences and their effect on exacerbating popular bias by presenting a holistic approach that integrates explainability into recommender systems, demonstrating its effectiveness in mitigating popularity bias. We enlist our contributions as follows:

- We propose a novel explanation-based perspective for recommendation, where both positive (pros) and negative (cons) explanations are incorporated to model user–item relationships, offering a more complete and fine-grained view of user preferences.

- We design a two-stage debiasing framework, **Expl-Debias**, that first learns base latent user–item preferences from interaction data (Stage-1), and then enhances the learning through contrastive training on explanation embeddings (Stage-2) to explicitly align relevance learning by comparing positive aspects with respect to the negative aspects for any item according to each user.
- We present an LLM-driven explanation generation and encoding pipeline that automatically constructs user-specific positive and negative explanations from textual profiles, enabling the model to capture true preferences and substantially reduce the dominance of popular items in top-K recommendations.

The remaining paper is organized as follows: Related work is discussed in Section 2. We detail our framework's methodology and design in Sections 3 and 4 respectively and offer the experimental setup in Section 5. We provide result analysis and conclusions in Sections 6 and 7.

2 Related Work

2.1 Explanations into Recommendation

Prior work has mainly used explanations to help consumers understand why recommended items match their preferences Chen et al. (2022b); Zhang and Chen (2020); Wardatzky et al. (2025), and as tools for fairness diagnostics in AI and recommender systems, including counterfactual reasoning Chang et al. (2024); Chen et al. (2024); Li et al. (2024); Ge et al. (2022b); Deldjoo et al. (2021). With the rise of Large Language Models (LLMs), researchers have further leveraged rich semantic texts to enhance user and item representations Silva et al. (2024); Xu et al. (2025a); Yang et al. (2024); Wang et al. (2024). Empirical evidence shows that integrating detailed item descriptions Xu et al. (2025b); Ren et al. (2024); Pauw et al. (2022) or preference-reasoning texts Zhang et al. (2023c); Gao et al. (2023) can substantially strengthen model training. Accordingly, explanations have been incorporated into objectives Sun et al. (2020); Dong et al. (2023) or training data Yu et al. (2024); Bismay et al. (2024) to tightly couple explanation quality with recommendation performance.
Recent work has explored mitigating popularity bias through LLM-generated profile texts Xv et al. (2022); Tang et al. (2024a); Liu et al. (2023); Wang et al. (2023), but these representations remain indirect and do not encode users' explicit likes and dislikes. Although generic explanation texts have been used to address diversity or hate-bias concerns Lin et al. (2024a); Cai et al. (2022); Yang et al. (2021), little attention has been paid to explanations that precisely articulate item pros and cons and their alignment with users. Motivated by contrastive learning approaches that utilize both positive and negative explanations Wang et al. (2025); Lin et al. (2024b), we introduce a method that tackles item-side popularity bias by jointly exploiting positive and negative reasoning texts while improving recommendation accuracy.

2.2 Popularity Debiasing

Popularity bias occurs when the top recommendations provided by the users are mostly occupied by the items interacted by many users (Zhu et al., 2022; 2021), leading to an unfair representation of recommendations filled with mostly popular items over other items for any user. This problem occurs in the user modeling phase of the feedback loop (Chen et al., 2021) and causes the over-representation of certain items over the others by virtue of its increased concentration within the interaction history of all the users. Most of the approaches are training-based in resolving the problem which mostly rely on regularization (Abdollahpouri et al., 2017; 2019; Wasilewski and Hurley, 2016; Chen et al., 2020; 2022a). There are certain novel approaches in resolving popularity bias such as: adversarial-based (Krishnan et al., 2018), chronological adjustments (Ji et al., 2020; Zhu et al., 2021), causality-related (Wang et al., 2021; Zhang et al., 2021; Bonner and Vasile, 2018) and information-based approaches (Tang et al., 2024b; Chen et al., 2023). However, there are not many approaches that have considered improving the deeper modeling of user-item interactions based on their true relevance via explanations. Although some works have incorporated the aspect of reasoning into debiasing algorithms (Wei et al., 2021; Liu et al., 2024; Zhang et al., 2023a), they have not directly included explanations into the learning schema. In this study, we focus on such an explanation-based perspective for mitigating popularity bias.

3 Preliminary

In this section, we will describe the preliminaries and analyze the real-world feasibility of our study, by highlighting the necessity of designing our method towards mitigating popularity bias.

3.1 Popularity Bias in Recommenders

We define dataset D comprising of m users $U = \{u_1, u_2 \dots u_m\}$ and n items $V = \{v_1, v_2 \dots v_n\}$. For each item $v \in V$, we can arrange it in increasing order of the count of users who have interacted with it (e.g., clicking/reviewing) and identify the most popular item set V^P. Typically, recommenders learn to rank items based on the input information $X_{u,v}$ for any user-item pair (u, v) that is either based on numerical ID values or contextual information that can be presented via texts and images. Let us denote any recommender g that learns to predict the user-item matching score for any user-item pair (u, v) as $g_\Theta(X, u, v)$.

Typically, we observe that the user-item matching relies heavily on existing information X that describes either only about the user u (name, location, age) or the item v (title, description, brand). However, such contextual information lacks an effective representation of direct correlations between the user and item. In order to resolve this problem, we propose leveraging explanations E for learning to rank items.

We note that explanations can be leveraged to describe the pros and cons about any item. Positive explanations E_+ illustrate why a user might prefer or engage with a particular item, thus helping the recommender model g to better capture preference patterns. Conversely, negative explanations E_- remark the reasons for disinterest or dissatisfaction, enabling g to recognize and avoid recommending items that align poorly with user preferences. By integrating the two types of explanations, recommender can learn in a more effective and deeper manner, leading towards more nuanced recommendation scoring. We can predict top-K recommendation list R_u^K for each user $u \in U$ with Equation 1.

$$R_u^K = \arg\max_{v \in V}^{K} g_\Theta(E_+, E_-, X, u, v) \tag{1}$$

This approach inherently reduces the risk of overly promoting popular items $v \in V^P$ by guaranteeing more balanced exposure of niche or less-known items that align closely with individual user interests. In this study, we express the goal of debiasing recommender g with utility loss $\mathcal{L}_{\text{utility}}(\cdot;\cdot)$ for each sample in D with label y mathematically in Equation 2.

$$\underset{\Theta}{\text{minimize}} \quad \mathcal{L}_{\text{utility}}(g_\Theta(E_+, E_-, X, u, v), y), \quad \forall (u, v) \in D \tag{2}$$

Here, positive explanations increase the relevance score when their semantics describe item aspects that align with user preferences, while negative explanations reduce the score when they reveal the disliked aspects about the item. Although we do not impose an explicit popularity-fairness constraint, the model introduces an implicit debiasing effect through explanation-aware contrastive scoring through both the explanations. We describe this difference more in detail in Section 4.2 in Stage 2.

3.2 Real-World Motivation

Popularity bias in recommender systems is increasingly apparent in real-world platforms, where frequently interacted items disproportionately dominate user feeds, marginalizing less known yet potentially relevant items. Consider a major online platform like Yelp, where highly popular businesses such as sea-food restaurants consistently dominate user recommendations due to their unusually increased presence across interaction of many customers. As a result, equally capable but less popular restaurants, such as niche local restaurants or new cuisines, receive minimal visibility, thus restricting the choices of users and potentially decreasing overall satisfaction of customers. We can observe similar phenomenon occurring across business-related e-commerce platforms like Amazon/E-Bay.

This disparity caused by item-side popularity poses significant challenges. Firstly, it impairs the user experience by limiting diversity and hindering the discovery of novel and personally relevant items. Secondly, it disadvantages manufacturers of niche products, constraining their market reach and growth opportunities. In order to mitigate such scenarios, there is a requirement for researchers to ensure more relevant recommendations customized to each customer. Incorporating explainability

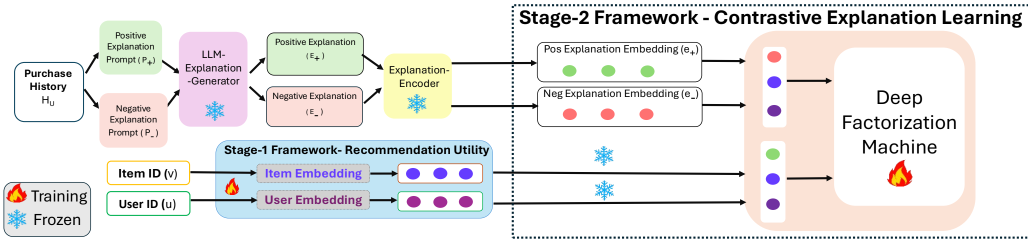


Figure 2: Our Expl-Debias Framework. Stage-1 consists of training user and item embeddings within a typical BPR-based pairwise learning style. Stage-2 consists of learning using a contrastive learning approach entirely based on positive and negative explanations across each user-item sample.

into recommendations can substantially resolve this issue. For instance, explicitly stating that a recommendation was made because "this niche smartphone matches your preference for advanced photography and long battery life" or that "this special cuisine aligns closely with your past visit history" helps users understand and trust these suggestions. Transparent explanations encourage users to explore and interact with less popular but personally relevant items, thereby fostering a fairer distribution of visibility.

In this study, we wish to present an even-handed perspective of explanations while modeling user interactions by providing both the pros and cons of any item according to each user.

4 Expl-Debias: Setup and Design

In this section, we discuss our debiasing framework named Expl-Debias that primarily consists of two training stages. Stage 1 involves training a recommender with basic ranking capabilities while Stage 2 involves a fine-grained approach that trains each sample to rank based on its explanation content for justifying the recommendation. Fig 2 represents our framework design. In Appendix A.4, we present a popularity-based ranking algorithm which performs final ranking by improving the unnecessary top ranking of popular items while maintaining the relevance of the recommender.

4.1 Stage 1: Vanilla Recommendation Utility

In order to endow fundamental recommendation utilities to any recommender, we optimize a learning to rank objective that involves supervised training of user-item samples with their actual interaction outcomes. This stage should enable any recommender to perform basic ranking utilities merely using ID-based inputs. At this stage, the model is trained to distinguish user-preferred items using only user and item IDs, without any auxiliary or explanation-based features. In order to ensure the model can learn strong preferences amongst items, we guarantee that the recommender possesses the capability of ranking one item over another for each user.

Therefore, we optimized our recommender using Bayesian Personalized Ranking (Rendle et al., 2012) loss function that learns to rank positive items $v_+ \in V$ (items that have been interacted/clicked: $(u, v_+) \in D$) higher than the negative items $v_- \in V$ (items that are not at all interacted by user: $(u, v_-) \notin D$). We express the optimization objective for recommender g with model parameters Θ in Equation 3.

$$\min_{\Theta} \sum_{(u,v_+)\in D;(u,v_-)\notin D} -\log(\sigma(g_\Theta(u, v_+)) - \sigma(g_\Theta(u, v_-))) \tag{3}$$

where σ denotes the sigmoid function. This learning-to-rank objective lays the foundation for more nuanced and fine-grained explanation-aware training in subsequent stages.

4.2 Stage 2: Contrastive Explanation Learning

In this stage, we increase the capabilities of the recommender to match user-item interactions based on enhanced inputs. While most of the traditional recommenders optimize to distinguish positive and negative items across all users, there is a lack of optimization in a more fine-grained approach. These recommenders lack the capability of completely understanding the contributing aspects of recommending an item to any user. We hypothesize that this information can be supplemented through explanations because they can be leveraged to describe the reasons for encouraging/discouraging any item. These explanations provide salient features that improve modeling the relevance of any item towards any user, thus offering increased advantages over existing traditional recommenders.

Therefore, we include explanations (say E) into the training objectives to offer higher scope of improvement in the recommendation quality.

However, explanations are typically generated post-hoc and in practice, most of the training pipelines do not include explanations into the model objectives. In order to resolve this issue, we first utilize the advanced capabilities of Large Language Models for generating explanations to explain why an item would possibly be/not be recommended to any user. While explanations are typically provided for encouraging item suggestion, we desire the true relevance of each user-item interaction. Therefore, we leverage LLM_{Generate} to generate two sets of explanations: positive which specify reasons why an item is recommended and negative which discourage by stating the possible disadvantages of the item. For generating the explanations, we use instruction prompts P_+ and P_- with the user/item information X found in the datasets. We describe the prompts and the context information used for generating explanations in Appendix A.1.

With the intention of enriching the generation of explanations, we supplement P_+ and P_- with the profiles of historical items $H_u = \{v_1^u, v_2^u, \dots\}$ that have already been interacted by the user as well as the current item that is being considered. We consider item titles and descriptions for creating item profiles, indicated as $\text{Prof}(\cdot)$, and we concatenate profiles for each item in H_u for forming the purchase history profiles. We formulate the mathematical expression for generating positive and negative explanations (E_+ and E_- respectively) as follows in Equations 4 and 5

$$E_+(u, v) = LLM_{\text{Generate}}(P_+ \oplus \text{Prof}(v) \oplus \{\text{Prof}(i)\ \forall i \in H_u\}) \tag{4}$$

$$E_-(u, v) = LLM_{\text{Generate}}(P_- \oplus \text{Prof}(v) \oplus \{\text{Prof}(i)\ \forall i \in H_u\}) \tag{5}$$

where $\oplus$ represents concatenation of texts according to the prompt format.

Following this step, we aim to train the recommender g to learn personalized preferences in a nuanced and even-handed manner. In order to achieve this, we intend to encourage the matching score of user-item interactions when provided with E_+ while discouraging the matching score when given E_-. We embed the textual explanations into embedding vectors using LLM_{Embed} for representing the explanations as inputs to recommender g. Through this approach, we aim to present a contrastive learning approach which matches with a higher probability when positive reasons are provided while lower probability scores are predicted when negative reasons are offered. In order to realize this goal, we again leverage BPR loss in a fine-grained level across each sample $(u, v) \in D$ for optimizing via a contrastive learning style as in Equation 6

$$\min_{\Theta} \sum_{(u,v)\in D} -\log\Big(\sigma\big(g_\Theta(u, v, e_+)\big) - \sigma\big(g_\Theta(u, v, e_-)\big)\Big) \tag{6}$$

where $e_+ = LLM_{\text{Embed}}(E_+)$ and $e_- = LLM_{\text{Embed}}(E_-)$.

Optimizing the contrastive objective in Equation 6 encourages to assign a larger value to $g_\Theta(u, v, e^+)$ when an item's positive explanation fits the user's preferences, and a smaller value to $g_\Theta(u, v, e^-)$ when the negative explanation highlights conflicting attributes. As a result, items whose positive and negative explanation scores diverge strongly are ranked higher, while items whose explanations provide weak alignment—often popular but irrelevant items—receive only a small separation between $g_\Theta(e^+)$ and $g_\Theta(e^-)$. Hence, contrastive explanation learning reduces the dominance of popular items. In order to verify the authenticity of the generated explanations, we evaluate them against the product reviews found in the datasets which is discussed in Appendix A.2. We additionally discuss the impact of Stage-2 on the model hidden layers in Appendix A.3.

5 Experiments

In this section, we provide experimental setup supplementing details regarding datasets, models, debiasing baseline methods and evaluation metrics. The training details are given in Appendix B.2.

5.1 Datasets

For this study, we chose e-commerce datasets since this domain has been previously studied for fairness works in literature and review-based explanations are prominent in this domain. We use Yelp business and Amazon product review based datasets such as Beauty and Sports. We preprocess the dataset such that each user and item has at least 5 reviews (5-core version). For

Stage 1, we follow the common leave-one-out protocol: the most recent interaction for each user is used as the test item, and the second most recent interaction is used for validation.[1]. For Stage 2, we use *exactly the same* train–validation–test split as in Stage 1. The contrastive explanation learning is performed solely on the training instances derived from the Stage 1 split, ensuring that no information leakage is introduced. Table 1 displays the dataset statistics.

Table 1: Dataset Statistics

Dataset	Users	Items	Reviews	Sparsity(%)
Beauty	22,363	12,101	198,502	0.0734
Yelp	30,431	20,033	316,354	0.0519
Sports	35,598	18,357	296,337	0.0454

5.2 Recommender Models

While many non-LLM debiasing approaches exist (e.g., graph-based or ID-based frameworks), these methods rely on user–item graphs or ID embeddings and are not directly applicable to text-centric LLM recommenders. We therefore focus on LLM-based models in this study.

- **TALLRec** (Bao et al., 2023): This model leverages a Low-Rank adaptation-based (LoRA) finetuning on LLaMa models. It only uses textual item data for recommendation tasks.
- **CoLLM** (Zhang et al., 2023b): This model combines both traditional IDs (Matrix Factorization for this study) and collaborative textual information by learning user and item embeddings along with finetuning LLaMa models.
- **LLaRA** (Liao et al., 2024): This model performs LoRA finetuning on large language models by enhancing item representation within textual prompts that include item embeddings from Matrix Factorization for all items along with text-based embeddings from LLaMa model.

5.3 Debiasing Baselines

To provide a competitive and fair comparison, we combine popular debiasing strategies with the above recommenders. The debiasing baselines are:

- **FairIPS** (Jiang et al., 2024): This in-processing debiasing method optimizes a weighted-loss that scores each sample based on the inverse popularity weight attached to the item's popularity group.
- **FairPrompt** (Xu et al., 2024): This prompting-based method evaluates all trained models with a unique fairness prompt that induces a much fairer recommendations by prompting them.

5.4 Metrics

We select standard recommendation metrics such as Normalized Discounted Cumulative Gain (NDCG) and HitRate (HR) that focus on the ranking accuracy of each model. For measuring debiasing, we use metrics that track the presence of popular items across recommendations amongst users. We discuss the debiasing metrics as follows and provide mathematical expressions in Appendix B.1.

- **Popularity Rate (PopRate):** The proportion of popular items amongst all the items across each user's top-K list.
- **Kullback Leiber Divergence (KLD):** The distributional divergence between the popular-niche item group distribution across the overall sample population $D_{true} = \{\frac{|V^P|}{V}, \frac{|V|-|V^P|}{V}\}$ and the predicted item group distribution D_{pred} in top-K lists.
- **User Popular-item Coverage (UPC):** The ratio of user count who have at least one popular item $v \in V^P$ recommended in their top-K lists to the total number of users.

6 Results

In this section, we discuss how Expl-Debias can improve recommendation performance while effectively controlling the popularity bias after Stages 1 and 2. We also analyze the effects of positive and negative explanations on the user preferences on popular and niche items qualitatively and quantitively. Additionally, we present ablation studies on using different explanation generators and encoders in Appendices C.2 and C.3, and our re-ranking algorithm results in Appendix C.4.

[1]This confirms that our test set reflects natural user behavior and is not constructed to contain a disproportionate number of popular items.

6.1 DIFFERENT TRAINING STAGES

Table 2: Performance and fairness of all baselines on Beauty, for $K = 3, 5, 10$. Best results per metric and K in **bold** while second-best results are underlined. ↑ means higher scores are better while ↓ means lower scores are better. Our framework improvements against the best baseline in each case are statistically significant (paired t-test and Wilcoxon signed-rank test, $p < 0.05$).

Method	$K = 3$					$K = 5$					$K = 10$				
	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)
TallREC	0.0991	0.1239	0.5115	0.5365	0.8759	0.1182	0.1705	0.4599	0.4260	0.9485	0.1469	0.2599	0.3839	0.2829	0.9888
CoLLM	0.1574	0.2026	0.7037	1.0439	0.9549	0.1852	0.2702	0.6351	0.8446	0.9825	0.2202	0.3785	0.5121	0.5376	0.9953
LLARA	**0.2184**	**0.2761**	0.6610	0.9173	0.9261	**0.2502**	**0.3535**	0.5594	0.6484	0.9541	**0.2861**	**0.4643**	0.4004	0.3121	0.9803
TallREC-FairIPS	0.0869	0.1094	0.5328	0.5851	0.8922	0.1025	0.1476	0.4884	0.4856	0.9573	0.1296	0.2323	0.4079	0.3256	0.9899
CoLLM-FairIPS	0.1501	0.1922	0.7417	1.1638	0.9718	0.1763	0.2560	0.6639	0.9259	0.9896	0.2114	0.3646	0.5277	0.5733	0.9979
LLARA-FairIPS	0.1787	0.2127	0.6665	0.9333	0.9325	0.2063	0.2799	0.5662	0.6650	0.9614	0.2821	0.3922	0.4070	0.3238	0.9830
TallREC-FairPrompt	0.1079	0.1295	0.4473	0.4005	0.8156	0.1237	0.1681	0.3927	0.2983	0.8987	0.1476	0.2428	0.3259	0.1902	0.9655
CoLLM-FairPrompt	0.1136	0.1511	0.6022	0.7565	0.9238	0.1399	0.2155	0.5661	0.6649	0.9762	0.1756	0.3261	0.4844	0.4770	0.9964
LLARA-FairPrompt	0.1650	0.2137	0.4513	0.4087	0.8207	0.1946	0.2857	0.4135	0.3359	0.9202	0.2290	0.4591	0.3419	0.2144	0.9800
Stage-1	0.1755	0.1875	0.4257	0.3588	0.7610	0.1863	0.2139	0.3858	0.2862	0.8552	0.2044	0.2708	0.3184	0.1793	0.9458
Stage-2	0.2030	0.2057	**0.3747**	**0.2672**	**0.7253**	0.2071	0.2160	**0.3552**	**0.2353**	**0.8460**	0.2191	0.2537	**0.3150**	**0.1745**	**0.9436**

Table 3: Performance and fairness of all baselines on Yelp for $K = 3, 5, 10$. Other details are the same as in Table 2.

Method	$K = 3$					$K = 5$					$K = 10$				
	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)
TallREC	0.2948	0.3232	0.4601	0.4263	0.7987	0.3228	0.3915	0.4027	0.3161	0.8735	0.3648	0.5222	0.3327	0.2004	0.9352
CoLLM	0.2118	0.2602	0.5666	0.6660	0.8859	0.2416	0.3329	0.4948	0.4996	0.9379	0.2806	0.4540	0.3818	0.2794	0.9768
LLARA	0.3109	**0.3875**	0.5861	0.7149	0.8405	0.3563	**0.4982**	0.4831	0.4743	0.8624	**0.4087**	**0.6602**	0.3180	0.1787	0.9316
TallREC-FairIPS	0.3277	0.3448	0.4106	0.3306	0.7539	0.3472	0.3926	0.3703	0.2599	0.8439	0.3698	0.5000	0.3071	0.1633	0.9251
CoLLM-FairIPS	0.2035	0.2502	0.5081	0.5287	0.8456	0.2317	0.3190	0.4412	0.3885	0.9136	0.2710	0.4413	0.3441	0.2178	0.9673
LLARA-FairIPS	0.3074	0.3840	0.5893	0.7233	0.8411	0.3533	0.4958	0.4871	0.4829	0.8638	0.4060	0.6586	0.3225	0.1853	0.8878
TallREC-FairPrompt	0.2145	0.2584	0.4972	0.5048	0.7749	0.2427	0.3270	0.4550	0.4160	0.8456	0.2809	0.4456	0.3754	0.2685	0.9197
CoLLM-FairPrompt	0.1167	0.1504	0.4778	0.4631	0.8440	0.1375	0.2009	0.4110	0.3313	0.9175	0.1687	0.2982	0.3223	0.1850	0.9768
LLARA-FairPrompt	0.2091	0.2725	0.4759	0.4592	0.8002	0.2508	0.3742	0.4128	0.3345	0.8670	0.3019	0.5322	0.3127	0.1713	0.9299
Stage-1	0.3088	0.3118	0.3909	0.2951	**0.7237**	0.3158	0.3290	0.3710	0.2612	**0.8383**	0.3372	0.3967	0.3204	0.1822	**0.8852**
Stage-2	**0.3565**	0.3581	**0.3839**	**0.2829**	0.7430	**0.3593**	0.3649	**0.3653**	**0.2516**	0.8622	0.3816	0.3983	**0.2982**	**0.1513**	0.9551

Table 4: Performance and fairness of all baselines on Sports for $K = 3, 5, 10$. Other details are the same as in Table 2.

Method	$K = 3$					$K = 5$					$K = 10$				
	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)
TallREC	0.0353	0.0482	0.2619	0.1058	0.5974	0.0466	0.0759	0.2372	0.0787	0.7383	0.0664	0.1378	0.2034	0.0473	0.8956
CoLLM	0.0981	0.1273	0.7723	1.2662	0.9818	0.1177	0.1753	0.6780	0.9670	0.9938	0.1449	0.2287	0.5154	0.5453	0.9985
LLARA	0.1371	0.1749	0.5736	0.6836	0.9016	0.1614	**0.2340**	0.4841	0.4765	0.9495	0.1968	**0.3444**	0.3658	0.2526	0.9844
TallREC-FairIPS	0.0402	0.0539	0.3183	0.1793	0.6764	0.0512	0.0807	0.2816	0.1297	0.8048	0.0733	0.1499	0.2354	0.0770	0.9298
CoLLM-FairIPS	0.0963	0.1244	0.6269	0.8225	0.9270	0.1138	0.1673	0.5511	0.6284	0.9673	0.1387	0.2445	0.4316	0.3702	0.9915
LLARA-FairIPS	0.0978	0.1257	0.7063	1.0520	0.9645	0.1151	0.1679	0.6030	0.7586	0.9839	0.1393	0.2430	0.4492	0.4045	0.9957
TallREC-FairPrompt	0.0434	0.0548	0.1693	0.0226	0.4271	0.0536	0.0795	0.1642	0.0196	0.5918	0.0727	0.1393	0.1542	0.0143	0.8125
CoLLM-FairPrompt	0.0962	0.1250	0.7584	1.2192	0.9784	0.1155	0.1720	0.6637	0.9253	0.9924	0.1427	0.2567	0.5001	0.5113	0.9981
LLARA-FairPrompt	0.0671	0.0893	0.3143	0.1736	0.6676	0.0825	0.1269	0.2793	0.1268	0.7943	0.1090	0.2094	0.2327	0.0742	0.9205
Stage-1	0.1845	0.1860	0.4318	0.3705	0.7819	0.1879	0.1946	0.4122	0.3336	0.8888	0.2014	0.2372	0.3645	0.2504	0.9653
Stage-2	**0.2139**	**0.2142**	**0.1214**	**0.0024**	**0.3198**	**0.2147**	0.2160	**0.1136**	**0.0010**	**0.4478**	**0.2187**	0.2600	**0.1072**	**0.0003**	**0.6648**

6.1.1 STAGE 1: VANILLA RECOMMENDATION UTILITY

We observe that traditional BPR-style vanilla recommendation training achieves a recommendation performance that is broadly comparable to existing baselines. This trend is consistent across all three datasets, as shown in Tables 2, 3, and 4. Stage-1 establishes strong fundamental recommendation capabilities by learning implicit user–item preferences through basic ID-based embeddings. Pairwise loss optimization induces strong relative ranking capabilities, which is reflected in consistently high NDCG scores. However, HR values are not superior to those of other baselines across datasets, since BPR primarily emphasizes the ordering of positive items over negatives and does not directly optimize for maximizing the absolute presence of relevant items within the top-K lists.

Despite these advantages, Stage-1 exhibits limitations in mitigating popularity bias, particularly in sparse datasets such as **Sports** (Table 4). For example, the PopRate@5 of 0.4122 remains high compared to LLARA-FairPrompt (0.2793), even though Stage-1 achieves better NDCG scores. Another observation is that Stage-1 is less competitive than more sophisticated LLM-based recommenders such as LLARA on dense datasets like **Beauty** and **Yelp**, despite displaying stronger fairness metrics. This difference can be attributed to LLARA's design of incorporating item embeddings from the user's interaction history directly into prompts, which enriches contextual learning compared to the simpler ID-based Stage-1 training. Therefore, Stage-1 performs comparably to all the baselines across datasets with respect to both recommendation and item debiasing, without leveraging explanations. Nonetheless, it does not fully alleviate popularity bias, as evidenced by the persistent presence of popular items in top-K lists (e.g., Sports at $K = 5$). These observations motivate the need for Stage-2 training, where explicit explanation-based preferences are integrated to achieve a stronger balance between recommendation performance and debiasing popularity bias.

6.1.2 STAGE 2: CONTRASTIVE EXPLANATION TRAINING

Stage-2 training introduces positive and negative explanations, encoded as embeddings and optimized using contrastive learning within the framework. Incorporating these contrastive explanation signals significantly boosts both recommendation quality and item-side fairness. As shown in Tables 2, 3, and 4, Stage-2 consistently improves ranking quality (higher NDCG and HR) while simultaneously reducing popularity bias (lower PopRate, KLD, and UPC), outperforming all baselines. Importantly, Stage-2 also outperforms Stage-1 across all datasets. For example, on **Sports**, Stage-2 raises NDCG@5 from 0.1879 to 0.2147 while sharply reducing PopRate@5 from 0.4122 to 0.1136. Similar trends hold for **Beauty** and **Yelp**, where improvements in NDCG are paired with consistent reductions in KL Divergence and User Popular-item Coverage. Notably, Stage-2 always lowers UPC, showing that the presence of at least one popular item in users' top-K recommendations is significantly reduced.

These results confirm that our Expl-Debias framework achieves significant improvements through the introduction of contrastive explanation-based training. The generated explanations enhance recommendation performance by explicitly revealing true user preferences, highlighting both likes and dislikes. Stage-2 training effectively captures fine-grained preferences by contrasting the pros and cons of each item for a given user. In this process, positive explanations align with aspects that users favor, thereby emphasizing item relevance, while negative explanations highlight unfavorable aspects, allowing the model to better account for irrelevance.

Our framework also mitigates the negative effects of popularity bias in recommendation lists. This advantage stems from Stage-2 training, which enables the recommender to both *promote niche items* aligned with user-specific pros and *demote popular but mismatched items* associated with user-specific cons through the positive and negative explanations respectively. As a result, the framework not only improves the ranking performance but also enforces debiasing constraints, as reflected by the consistently lower UPC values across datasets. Similar to Stage-1, NDCG exhibits larger improvements than those in HR, which can be attributed to the pairwise loss optimization that prioritizes relative ranking quality of items. In the meantime, HR also improves under Stage-2 and in some cases performs better than all baselines, indicating that explanation-aware learning ensures better recommendation performance. Overall, explanation-aware Stage-2 integrates explicit user preferences derived from explanations with the implicit preferences learned during ID-based Stage-1 training. Therefore, Expl-Debias framework offers an empirically effective mechanism in providing a principled approach to mitigating popularity bias without any major sacrifice towards the recommendation performance.

6.2 Effect of Positive/Negative Explanations

In this section, we discuss the direct impact of positive and negative explanations in order to visualize. our framework's effectiveness in mitigating popularity bias. We focus this study towards analyzing how positive explanations can promote in Table 5. We selected a random 1% subset of items that lie in the bottom 10 percentile of item popularity (which we term as niche as they receive very little exposure). Due to space limitation, analysis of negative explanations is given in Appendix C.1

Table 5: Effect of inducing positive explanation embeddings on a random subset of niche items in Beauty. N-NDCG and N-HR denote the ranking scores of niche items in top-5 recommendations. Mean Inverse Rank (MIR) is the average reciprocal rank of each niche item across users, and Avg. Probability is the mean recommendation probability of a niche item. Blue indicates promotion (higher probability and metric scores) compared to the no-explanations setting.

Setting	N-NDCG@5	N-HR@5	MIR	Avg. Probability
No Explanations	0.0221	0.0383	0.0451	0.4330
Positive Explanations	0.0457	0.0831	0.0883	0.9348
Improvement (in %)	+106.79 ↑	+130.03 ↑	+95.79 ↑	+115.89 ↑

6.2.1 Effect of Positive Explanations

In Table 5, we can observe that positive explanations are quite consistent in ensuring an overall promotion towards increasing the presence of arbitrary niche items amongst top-5 recommendations for each user within the **Beauty** dataset. Higher N-NDCG and N-HR scores manifest the fact that the chosen subset of niche items are ranked higher and found more frequently amongst the top-5 ranked lists whenever positive explanations are introduced for predictions in comparison to the no-explanation setting. Similarly, average probability and MIR increase by over 100%, indicating that niche items are both ranked higher and assigned substantially larger user-item probability scores. These results confirm that positive explanations are highly effective in *promoting niche items*, as they identify fine-grained relevant reasons that establish why any user truly prefers an item. Niche items typically suffer from limited reach due to their minimal presence across historical user-item interactions. However, by introducing positive explanations that explicitly include user-aligned positive aspects that specify why a user prefers an item (e.g., highlighting beneficial features or attributes), the model boosts their presence in the top-5 lists since they can identify the hidden true relevance between any user and any niche item. As a result, positive explanations increase the relevance for niche items, thereby *promoting them*, leading to improved ranking position and recommendation probability.

6.3 CASE STUDY

Table 6: Case study showing how Stage-2 training promotes a niche item (just as in Section 6.2.1 and demotes a popular item after being trained from Stage-1 for user ID AC1KIJ6OYGVSK in **Beauty**. Blue indicates explicit reveal of **positive aspects** while Red reveals **negative aspects** of the product.

Item ID & Title	Rank Shift	Generated Explanations	User-Written Review Snippets
 **B001KYRMBU (Niche Product)** *L'Oreal Le Kohl Pencil Smooth Defining Eyeliner*	**6 → 1** ↑ 5	**Positive:** . . . will purchase because consumer is looking for a pencil eyeliner that would provide a smooth, precise application on the skin. **Negative:** . . . will not purchase because this product is chemical eyeliner but consumer is looking for environmental-friendly products.	**Positive Aspects:** "My skin feels even smoother and I swear my spots are starting to diminish.", "It also leaves your skin feeling velvety smooth" **Negative Aspects:** "I HATE how it smells. It has a weird Neutrogena glycerin soap bar/plastic/vitamin odor that I can't stand."
 **B0018S8MZ8 (Popular Product)** *Clean & Clear Blackhead Eraser Kit*	**2 → 19** ↓ 17	**Positive:** . . . will purchase because this product will induce relief in removing blackheads and address blackheads into cleansing routine. **Negative:** . . . will not purchase because this product will raise hyper-pigmentation as side-effect.	**Positive Aspects:** "pores relaxed a little bit","I noticed this is a great pimple-zapper." **Negative Aspects:** "it's just the discoloration I really want to change."

In this section, we provide a real-world scenario by analyzing how well the generated explanations are aligned with the user reviews and how their effect can be observed across the ranking shift when transitioning from Stage-1 to Stage-2. We can observe in Table 6 a consumer in the Beauty dataset with user ID *AC1KIJ6OYGVSK*. The first row product L'Oreal eyeliner is not quite popular in the Beauty dataset, but the product has been ranker higher to the 1st rank into the top-5 list from Stage-1 (the 6th rank). We can notice that the smooth skin requirements mentioned by the user regarding was aligned directly through the positive explanations (repeated skin and smooth words) while negative explanations mention regarding dislike towards chemical, but does not provide a specific match to their reviews regarding their hate for glycerin. On the contrary, the second row shows a popular item such as the Blackhead eraser kit, which is strongly demoted from the top-2 rank into the a much lower rank outside the immediate consumer visibility. In this case, the negative explanation directly matches with the user review snippets that reveal the skin discoloration concerns (hyperpigmentation reference). Additionally, the positive explanations do not reveal much beyond relaxed skin pores, which does not mention about the relief offered by the kit. From both examples, we can observe how closely both the positive and negative explanations contribute to the recommendation abilities of the model and justify the preferences of the consumer.

7 CONCLUSION

In this work, we propose **Expl-Debias**, a recommendation framework that incorporates explanation-aware training to improve both recommendation and debiasing performance. Our framework solves the existing problem of balancing recommendation accuracy along with controlling popularity bias in order to enhance user satisfaction. Our two-stage training design incudes Stage-1 which establishes fundamental learning of strong recommendation utilities, and Stage-2 which leverages contrastive learning on each user-item sample. Stage-2 training contrasts positive and negative explanations to promote niche items and demote irrelevant popular items, and reduce overall popularity bias. We have empirically validated that our framework is effective in maintaining strong recommendation performance while also maintaining low popularity item presence. We discussed the qualitative and quantitative aspects of our contrastive explanation learning approach towards recommendation and debiasing performance.

Admittedly, our framework also possesses certain limitations. It currently relies on text-only explanations without considering the impact of multi-modal data such as images/videos etc. Additionally, other characteristics of recommendation fairness such as user-side fairness and how effective our framework is towards resolving such problems are yet to be answered. Future work will focus on scaling our solution towards real-world pipelines where mitigating popularity bias along with maintaining recommendation is critical.

A Methodology

A.1 Explanation Generation Prompts

Positive Explanation Prompt (P_+): Given the profiles of the purchasing history of this consumer, can you provide a reason for why this consumer will purchase the current product?

Answer with one sentence with the following format: "The consumer will purchase this product because ..."

Profiles of Purchasing History: $<$ Purchase-History-Profiles $>$
Current Product Profile: $<$ Current-Item-Profile $>$

Negative Explanation Prompt (P_-): Given the profiles of the purchasing history of this consumer, can you provide a reason for why this consumer will not purchase the current product?

Answer with one sentence with the following format: "The consumer will not purchase this product because ..."

Profiles of Purchasing History: $<$ Purchase-History-Profiles $>$
Current Product Profile: $<$ Current-Item-Profile $>$

User feedback imbalance and applicability to low-history users. The proposed framework does not rely on observed positive and negative interactions to construct the explanation pair (e^+, e^-). Instead, both explanations are *generated* for every (u, v) pair using the same item context and the user's available review history, regardless of its size. This ensures that each user receives a balanced positive–negative explanation pair, even when their observed feedback is highly asymmetric or limited.

Because the explanations are conditioned primarily on the purchase history of the user, there is no additional external information that causes a skew while generating the explanations. For generating the explanations, we supplement the item profiles through their titles and descriptions, which provides sufficient information regarding user interests towards items. In practice, we observe that users with minimal history still obtain meaningful explanation signals: the positive explanation emphasizes attributes consistent with the few known user preferences, while the negative explanation highlights the disliked aspects. Thus, we conclude that we provide same information as context and we confirm that this approach does not introduce imbalance between the positive and negative explanation generated. During Stage-2, our framework only needs the relative alignment between these two signals which provide an even-handed representation of the user's likes and dislikes.

A.2 Fidelity of LLM-generated Explanations

Table 7: Evaluation of the positive explanations generated by our LLM model used for Stage-2 training. We evaluate all the positive explanations generated against the original user-written product reviews found in the dataset. We report the macro-average Precision (P), Recall (R) and F1 (F1) of BERTScore metrics along with the standard deviation reported in ($\pm$). Our results are statistically significant with 95% confidence reported.

Dataset	BERTScore-P (in %)	BERTScore-R (in %)	BERTScore-F1 (in %)
Beauty	84.25±1.89	83.17±2.19	83.68±1.64
Yelp	84.48±1.39	82.03±2.06	83.22±1.41
Sports	84.19±1.76	83.07±2.17	83.61±1.57

Across all three datasets, the positive explanations generated by our LLM exhibit consistently strong semantic fidelity to the ground-truth user reviews, as shown in Table 7. We evaluate each explanation against the corresponding user-written review using BERTScore (Zhang et al., 2020), which computes token-level semantic similarity via contextualized RoBERTa embeddings. This allows us to assess whether an explanation is grounded in the review text without requiring lexical overlap. The results demonstrate high factual alignment: BERTScore Precision averages around 84%, indicating that most tokens in each generated explanation are supported by content present in the user's review, thus avoiding hallucinated or spurious information. Recall remains similarly strong (82–83%), showing that the explanations capture the majority of salient aspects mentioned in the review, despite being considerably shorter (one–two sentences). The balanced macro-averaged F1 scores (83–84%) further confirm that explanations are both specific and comprehensive.

Table 8: Evaluation of the negative explanations generated by our LLM model used for Stage-2 training. We evaluate all the negative explanations generated against the original user-written product reviews found in the dataset. Similar description can be found in Table 7. Our results are statistically significant with 95% confidence reported.

Dataset	BERTScore-P (in %)	BERTScore-R (in %)	BERTScore-F1 (in %)
Beauty	83.75±1.74	82.71±2.14	83.21±1.56
Yelp	83.69±1.23	81.61±2.00	82.63±1.33
Sports	83.57±1.61	82.68±2.11	83.11±1.49

We observe a comparable pattern for negative explanations (Table 8). These counterfactual explanations also exhibit high semantic fidelity (F1 scores of 82–83%), demonstrating that they do not introduce information absent from the review and thus do not inject noise into the contrastive supervision used in Stage 2. Finally, the standard deviations across all metrics are small (±1.3–2.1), reported with 95% confidence interval, indicating that explanation quality is highly stable. Therefore, these results confirm that our LLM produces faithful and reliable positive and negative explanations, providing high-quality training data for the contrastive debiasing procedure in Stage 2.

A.3 DEEPFM INTERNAL REPRESENTATIONS AFTER STAGE-2

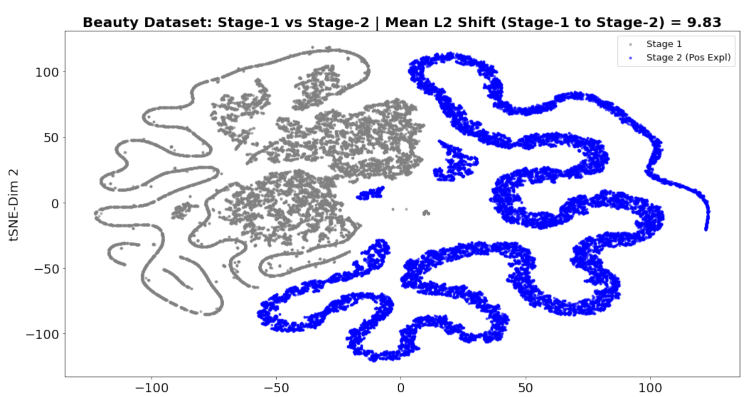


Figure 3: Visualization of DeepFM model representations using t-SNE to compare the effect of embedding changes between Stage-1 and Stage-2 on Beauty dataset. We visualize the DeepFM hidden layer representation after Stage-1 and Stage-2 and we compare the difference between Stage-1 and Stage-2 training.

Fig. 3 shows a t-SNE projection of the contextual hidden layer representations of DeepFM for the Beauty dataset across Stages 1 and 2. Stage-1 embeddings (in gray), form a dense and largely unstructured cluster region. We can observe that the cluster is unstructured in general, indicating that the model encodes minimal organization due to the simplistic training strategy followed during this stage. This behavior is directly representative of the straight-forward BPR ranking objective offered by Stage-1 which merely considers ID-based information for training and it does not guarantee an effective balance in reducing the presence of unnecessary popular items.

However, Stage 2 embeddings (in blue), trained under the more sophisticated contrastive objective, exhibit a different geometric pattern. For Stage 2, the blue cluster is much smoother and it represents coherent manifolds with clear global structure and occupy a region that is well separated from the Stage 1 cluster. This reorganization provides direct visual evidence that Stage 2 modifies the underlying representation space drastically. We can discuss this behavior with respect to the explanation-aware training objective, which leads DeepFM to align more closely with the user likes and dislikes, thus leading to more comprehensive understanding of the user-item relationships, which ensures a better trade-off between the debiasing and recommendation utilities.

A.4 Popularity-aware Ranking

Algorithm 1 Popularity-constrained Ranking

Require: For each user u: candidate items V^u, popular items V^P, Candidate α values $\mathcal{A} \subset [0, 1]$, Top-$K$, popularity constraint τ.

for each user $u \in U$ **do**
  Initialize $bestNDCG \leftarrow -\infty$, optimal $\alpha_u^* \leftarrow 0$
  for each $\alpha \in \mathcal{A}$ **do**
    for each $v \in V^u$ **do**
      $s_{\text{pos}}^{(u,v)} \leftarrow \sigma(g_\Theta(u, v, e_+))$
      $s_{\text{neg}}^{(u,v)} \leftarrow \sigma(g_\Theta(u, v, e_-))$
      $s_{\text{zero}}^{(u,v)} \leftarrow \sigma(g_\Theta(u, v, e = 0))$
      $S_{(u,v)}(\alpha) \leftarrow \alpha \cdot (s_{\text{pos}}^{(u,v)} - s_{\text{neg}}^{(u,v)}) + (1 - \alpha) \cdot s_{\text{zero}}^{(u,v)}$
    end for
    $R_u^K(\alpha) \leftarrow$ Top-K items ranked based on $S_{u,v}(\alpha)$ scores.
    $\text{NDCG}_u(\alpha) \leftarrow \text{NDCG@}K$ for $R_u^K(\alpha)$:
    $\text{PopRate}_u(\alpha) \leftarrow \frac{|\{v \in R_u^K(\alpha) \cap V^P\}|}{K}$
    if $\text{PopRate}_u(\alpha) \leq \tau$ **then**
      if $\text{NDCG}_u(\alpha) > bestNDCG$ **then**
        $bestNDCG \leftarrow \text{NDCG}_u(\alpha)$
        $\alpha_u^* \leftarrow \alpha$
      end if
    end if
  end for
  if $bestNDCG = -\infty$ **then**
    Select α_u^* with highest NDCG@K (ignore constraint)
  end if
  Output $R_u^K(\alpha_u^*)$ as Top-K ranking for user u
end for

Following the two-stage training paradigm, we desire to achieve an enhanced ranking mechanism that scores items by balancing the positive reasons and negative reasons when recommending an item. For each user-item pair, we present an even balancing between explanation-informed preferences and generic-attribute based preferences. Therefore, it is important to identify an optimal balance across these terms by considering both the utility significance and popularity bias constraints. This final step in our Expl-Debias framework addresses the well-documented challenge of popularity bias by designing a post-hoc re-ranking mechanism, wherein the influence of contrastive explanations is adaptively controlled at the level of each individual user. Our objective is to maximize recommendation relevance while ensuring the presence of popular items within a controllable range for every user.

To this end, we design a holistic ranking approach leveraging the fact that our model is capable of learning both general utility without any explanation and enhanced user-item relevance derived from contrastive explanation embeddings. For each user u and candidate item v, we leverage three types of scores obtained from the recommender g_Θ:

- $s_{pos}^{(u,v)}$: Score from the positive explanation embedding e_+.
- $s_{neg}^{(u,v)}$: Score from the negative explanation embedding e_-.
- $s_{zero}^{(u,v)}$: Score from the zero explanation embedding[2].

We propose a simple α-weighted linear combination to generate the final ranking score:

$$S_{u,v}(\alpha_u) = \alpha_u \cdot (s_{pos}^{(u,v)} - s_{neg}^{(u,v)}) + (1 - \alpha_u) \cdot s_{zero}^{(u,v)}, \tag{7}$$

where $\alpha_u \in [0, 1]$. The difference between the positive and negative explanation based scores provides a qualitative differential score for any user-item pair[3] modeling regarding why an item should be preferred and why it should not, thus mitigating the confounding effect of item popularity. In order to realize the original utility based score, s_{zero} term retains the general ranking functionalities learned from interaction data. However, there exists a challenge in determining the exact weight ratio between the two terms in Equation 7 because it varies depending on each user's specific preferences. Therefore, we design an algorithm that automatically computes the optimal α_u^* by maximizing the ranking relevance measured by NDCG for top-K recommendation lists. For each computed list, we also ensure a popularity-based constraint $\tau \in [0, 1]$ on the ranking lists to enforce the item fairness constraints. We present our popularity-aware ranking algorithm in Algorithm 1.

B Experimental Setup

B.1 Debiasing metric formulas

$$PopRate@K = \frac{\sum_{u \in U} \sum_{v \in R_u^K} \mathbb{1}(v \in V^P)}{K \cdot m}$$

$$KLD(D_{pred} \| D_{true}) = \sum_{i \in 0,1} D_{pred}(i) \ln(\frac{D_{pred}(i)}{D_{true}(i)})$$

$$UPC@K = \frac{\sum_{u \in U} \mathbb{1}(v \in V^P \text{and } v \in R_u^K)}{m}$$

B.2 Training Details

We use Deep Factorization Machines (DeepFM) as the base recommender, LLaMa-3 models for explanation generation, and LLaMa-2-7B for encoding the explanations due to their open-source nature and resource efficiency.All models including the baselines are trained with a batch size of 16. We search hyperparameters such as learning rate in range $[1e^{-7}, 1e^{-6}, 1e^{-5}, 5e^{-4}, 1e^{-3}, 1e^{-2}]$ and weight decay in range $[1e^{-7}, 1e^{-6}, 1e^{-5}, 5e^{-4}, 1e^{-3}, 1e^{-2}]$ on the validation data for all DeepFM models and baselines. This ensures that all baselines are tuned as rigorously and under the same search space as our proposed model. We search embedding size x in $\{8, 16, 32, 64, 128, 256\}$ and set 3 hidden layers of sizes $[4x, 2x, x]$. TALLRec, CoLLM, and LLaRA are trained for up to 10 epochs with early stopping (patience of 3). FairIPS uses the same optimizer and search space and was also trained for 10 epochs, with number of item groups chosen between $[2, 5, 10]$, while FairPrompt applies a consistent fairness prompt at inference across all recommenders.
Stage-1 training runs for 50 epochs at most while Stage-2 training (using frozen user and item

[2]This can be achieved by introducing zero-padded explanations into LLM_{Embed}

[3]This difference score is not the same as observed in Stage 2, and we do not train any data in this re-ranking paradigm.

embeddings) takes 5 epochs at most. We perform early stopping with tolerance up to 3 consecutive epochs for all the models. Explanations are generated with a limit of 50 tokens and filtered for grammatical and semantic consistency through direct supervision. We evaluate Stage-1 models with pad-filled explanations, indicating zero explanations, and use positive explanations for Stage-2 evaluation[4]. We fine-tune all the models on four NVIDIA RTX A6000 GPUs. We provide our code at `https://anonymous.4open.science/r/Expl-Pop-Bias-089A/` for additional details.

C Results Analysis

C.1 Effect of Negative Explanations

Table 9: Effect of inducing negative explanation embeddings on a random subset of popular items in Beauty. P-NDCG and P-HR denote the ranking scores of popular items in top-5 recommendations. MIR and Avg. Probability are defined similar to Table 5. Red indicates demotion (lower probability and metric scores) compared to the no-explanations setting.

Setting	P-NDCG@5	P-HR@5	MIR	Avg. Probability
No Explanations	0.0974	0.1573	0.1225	0.7088
Negative Explanations	0.0005	0.0008	0.0183	0.1701
Decline (in %)	-99.49 ↓	-99.49 ↓	-85.06 ↓	-76.00 ↓

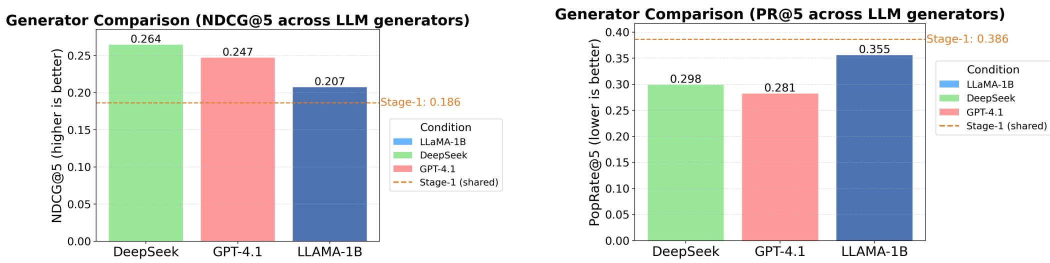
We discuss the effect of negative explanations in demoting a randomly chosen subset of 1% popular items in Table 9.

We observe that negative explanations exhibit a strong demotion effect upon a random subset of popular items amongst the top-5 recommendation lists for any user. This can be noticed through the steep decline (about 99.5% decline) of the P-NDCG and P-HR scores which indicate that popular items on average are pushed lower in ranking and their overall presence is also reduced amongst top-5 items. Lower MIR and Avg. Probability scores support our findings that the chosen subset of popular items is strongly demoted through the inclusion of negative explanations in comparison to the no-explanation setting. Popular items are often over-recommended due to their overwhelming presence in historical interaction data, leading to redundancy in recommendation lists. However, introducing negative explanations specifies why a user does not prefer an item by highlighting aspects or features that conflict with user preferences (e.g., features that the user may not prefer but the item possesses). Therefore, these explanation texts induce the true holistic perspective about popular items which refines the ranking process and reduces the undue dominance of popular items in the top-5 lists. In conclusion, we can infer that negative explanations reduce the unnecessary recommendations of popular items and effectively *demote them*.

C.2 Different Explanation Generators

In this section, we study the effect of varying explanation generation sources in terms of the quality of explanations and their impact on the model performance. We evaluate three different LLM generators: DeepSeek-7B, ChatGPT-4.1-mini and LLaMa-1B, with all sharing the same Stage-1 checkpoint (which does not involve explanations). In Fig. 4a, we notice that DeepSeek performs the best with respect to the recommendation performance (NDCG@5 as 0.264) followed by GPT (0.247) and then LLaMa (0.207). We can observe a similar trend of popularity bias mitigation in Fig. 4b where DeepSeek (PopRate@5 as 0.298) and ChatGPT (0.281) models display lower popularity rate scores in comparison to LLaMa (0.355). The superior performance of DeepSeek with respect to both aspects can be attributed to its larger size which offers high-quality explanation texts for training Stage-2. Similarly, ChatGPT is a close-sourced LLM that has trained on a larger corpus which includes human feedback, leading to more fluent and semantically rich completions and higher-quality explanations. They can offer a better view of user-item interactions and thus induce stronger alignment with the true user preferences. As a comparison, LLaMa-1B is smaller in size

[4]It is intuitive to perform top-K recommendation using positive reasons after Stage-2, evaluating based on what encourages an item to be recommended to any specific user.



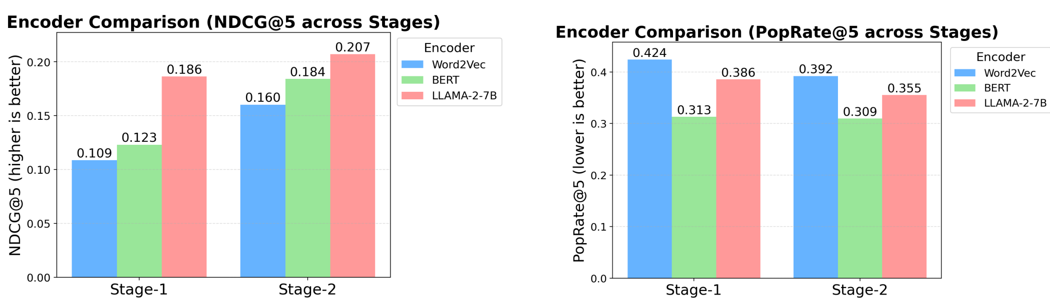
(a) NDCG@5 (b) PopRate@5

Figure 4: Explanation LLM Generators comparison — NDCG@5 and PopRate@5 across the recommender trained on Beauty with different LLM generations. Stage-1 is shared for all the scenarios since they do not utilize explanation texts during training.

($\sim$1B parameters) whose explanation generation capability may not be as advanced as ChatGPT or DeepSeek.

Despite the performance gains in terms of both the recommendation and debiasing aspects as depicted in Fig. 4, we still choose LLaMa as our default generator because of its efficiency advantages. With only $\sim$1B parameters, it provides much faster completions and lower memory cost compared to larger models such as DeepSeek-7B, making it suitable for large-scale experiments. Moreover, as an open-source model, LLaMA avoids request rate limits and latency constraints imposed by close-sourced APIs (as in the case of ChatGPT) which requires longer time for serving a large number of requests and are thus practically infeasible. While its explanation quality is sub-optimal, LLaMa still provides comparable generation capabilities and substantial improvements in speed and cost-efficiency, highlighting lesser demand of computational resources, which make LLaMa-1B a practical choice when balancing performance and efficiency.

C.3 Different Explanation Encoders



(a) NDCG@5 (b) PopRate@5

Figure 5: Encoder comparison — NDCG@5 and PopRate@5 across different encoders trained on Beauty dataset. Each encoder represents texts to different embedding sizes leading to difference in model architectures and thus each scenario requires separate training of both Stage-1 and Stage-2.

We study the ablation effect of different encoders used to transform textual explanations into numerical embeddings. Specifically, we compare **Word2Vec**, **BERT**, and **LLaMA-2-7B**, which differ in language modeling capabilities and embedding dimensionality. As shown in Fig. 5a, LLaMA consistently delivers the strongest recommendation accuracy across both stages, followed by BERT and then Word2Vec. This performance trend reflects the increasing representational power of the encoders: more advanced language modeling yields richer semantic representations of positive and negative explanation texts, leading to improved ranking quality. In contrast, BERT attains the lowest PopRate@5 values (Fig. 5b), owing to its bidirectional masked language modeling objective, which captures in-place contextual information within the explanation texts without overemphasizing upon unnecessary pre-trained knowledge. This allows BERT to represent all the items in an even-handed manner which contributes to reducing popularity bias. Word2Vec performs worst on both accuracy and fairness due to its simplistic encoding strategy which may not possess valuable information that reveals the user preferences. Another interesting observation is that Stage-2 training is consistently better in terms of recommendation and debiasing than Stage-1 training irrespective of any encoder being used.

Despite BERT's advantage in fairness, we adopt LLaMA as the default encoder because of its superior representational richness, driven by a larger embedding size (4096) compared to BERT (768) and Word2Vec (300). Although BERT models perform best in reducing popularity bias, they are not competent in retaining recommendation performance in comparison to LLaMa models. Our framework is designed to prioritize strong recommendation performance while constraining popularity bias within acceptable limits. Thus, we conclude that LLaMa offers the best tradeoff towards these two aspects, while BERT models sacrifice recommendation performance largely in comparison to their gain in debiasing capabilities.

C.4 Effect of Popularity aware Re-ranking

C.4.1 Improvements over Stage-1 and Stage-2 evaluation

Table 10: Performance and fairness comparing Stage-1, Stage-2 and our Re-Ranking algorithm, for $K = 5$ on Beauty. ↑ means higher scores are better while ↓ means lower scores are better. Best results per metric and K in **bold**.

Method	$K = 5$				
	NDCG (↑)	HR (↑)	PopRate (↓)	KLD (↓)	UPC (↓)
Stage-1	0.1863	0.2139	0.3858	0.2862	0.8552
Stage-2	0.2071	0.2160	0.3552	0.2353	0.8460
Re-Rank ($\tau = 1$)	**0.2604**	**0.3220**	**0.3439**	**0.2174**	**0.8199**

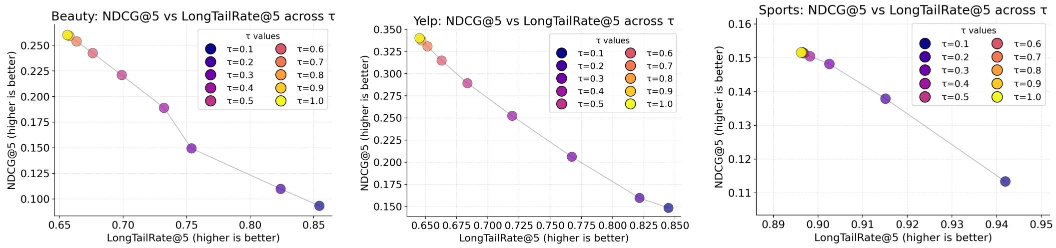
In Table 10, we can notice that our Re-Rank performs even better than our already effective Stage-2 based training, by offering a much better recommendation performance (larger NDCG and HR) while also offering lesser popular item presence via lower PopRate, KLD and UPC scores. This demonstrates that re-ranking can serve as an effective post-processing strategy that complements the explanation-aware training of Stage-2.

The advantage of our re-ranking approach stems from its design: the algorithm enforces stricter constraints on the inclusion of popular items within each user's top-5 recommendations, while greedily selecting the optimal weight factor (α) that maximizes NDCG whenever we are not able to satisfy the constraints. By jointly emphasizing ranking utility and debiasing constraints during re-scoring, the re-ranking step achieves a more favorable balance over mere stage-based training alone. These results highlight that explanation-aware training and fairness-oriented re-ranking are complementary—Stage-2 provides strong user–item preference signals, and re-ranking refines the final recommendation list to ensure both high recommendation performance and reduced popularity bias.

C.4.2 Ablation of Hyperparameters (τ)

We can observe that the re-ranking algorithm is effective in offering a reasonable trade-off between NDCG@5 and LongTailRate@5[5], offering consistent trends across all the datasets in Fig. 6. As we allow larger τ values, we can observe a visible reduction in the debiasing capabilities while there is an improvement in the recommendation performance (moving left and upwards trends for increasing τ in each plot). However, we can observe steeper trade-off curves across denser Beauty (in Fig. 6a) and Yelp (in Fig. 6b) datasets while Sports dataset (in Fig. 6c) exhibits a much more relaxed trade-off constraints especially across larger τ. Therefore, we can conclude that deciding τ depends on whether recommendation performance or fairness performance is preferred, with smaller τ yielding fairer but less accurate recommendation outcomes.

[5]It is defined as 1 - PopRate@5.



(a) Beauty (b) Yelp (c) Sports

Figure 6: Ablation TAU study

References

Himan Abdollahpouri. 2019. Popularity Bias in Ranking and Recommendation. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, Honolulu HI USA, 529–530. https://doi.org/10.1145/3306618.3314309

Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, Como Italy, 42–46. https://doi.org/10.1145/3109859.3109912

Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-ranking. (2019). arXiv:1901.07555 [cs.IR] https://arxiv.org/abs/1901.07555

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 *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. ACM. https://doi.org/10.1145/3604915.3608857

Millennium Bismay, Xiangjue Dong, and James Caverlee. 2024. ReasoningRec: Bridging Personalized Recommendations and Human-Interpretable Explanations through LLM Reasoning. arXiv:2410.23180 [cs.IR] https://arxiv.org/abs/2410.23180

Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 104–112. https://doi.org/10.1145/3240323.3240360

Yi Cai, Arthur Zimek, Gerhard Wunder, and Eirini Ntoutsi. 2022. Power of Explanations: Towards automatic debiasing in hate speech detection. arXiv:2209.09975 [cs.CL] https://arxiv.org/abs/2209.09975

Peter W Chang, Leor Fishman, and Seth Neel. 2024. Model Explanation Disparities as a Fairness Diagnostic. https://openreview.net/forum?id=4MvHiijJL3

Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. 2023. Unbiased Knowledge Distillation for Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, 976–984. https://doi.org/10.1145/3539597.3570477

Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2021. Bias and Debias in Recommender System: A Survey and Future Directions. arXiv:2010.03240 [cs.IR] https://arxiv.org/abs/2010.03240

Xu Chen, Yongfeng Zhang, and Ji-Rong Wen. 2022b. Measuring "Why" in Recommender Systems: a Comprehensive Survey on the Evaluation of Explainable Recommendation. arXiv:2202.06466 [cs.IR] https://arxiv.org/abs/2202.06466

Ziheng Chen, Jia Wang, Jun Zhuang, Abbavaram Gowtham Reddy, Fabrizio Silvestri, Jin Huang, Kaushiki Nag, Kun Kuang, Xin Ning, and Gabriele Tolomei. 2024. Debiasing Machine Unlearning with Counterfactual Examples. arXiv:2404.15760 [cs.LG] https://arxiv.org/abs/2404.15760

Zhihong Chen, Jiawei Wu, Chenliang Li, Jingxu Chen, Rong Xiao, and Binqiang Zhao. 2022a. Co-training Disentangled Domain Adaptation Network for Leveraging Popularity Bias in Recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid Spain, 60–69. https://doi.org/10.1145/3477495.3531952

Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative Domain Adaptation with Non-Displayed Items to Improve Long-Tail Performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 579–588. https://doi.org/10.1145/3397271.3401043 arXiv:2005.10545 [cs].

Yashar Deldjoo, Alejandro Bellogin, and Tommaso Di Noia. 2021. Explaining recommender systems fairness and accuracy through the lens of data characteristics. *Information Processing & Management* 58, 5 (2021), 102662.

Yashar Deldjoo, Dietmar Jannach, Alejandro Bellogin, Alessandro Difonzo, and Dario Zanzonelli. 2022. A Survey of Research on Fair Recommender Systems. http://arxiv.org/abs/2205.11127 arXiv:2205.11127 [cs].

Xue Dong, Xuemeng Song, Na Zheng, Yinwei Wei, and Zhongzhou Zhao. 2023. Dual Preference Distribution Learning for Item Recommendation. *ACM Transactions on Information Systems* 41, 3 (Feb. 2023), 1–22. https://doi.org/10.1145/3565798

Wenqi Fan, Han Xu, Wei Jin, Xiaorui Liu, Xianfeng Tang, Suhang Wang, Qing Li, Jiliang Tang, Jianping Wang, and Charu Aggarwal. 2023. Jointly Attacking Graph Neural Network and its Explanations. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, Anaheim, CA, USA, 654–667. https://doi.org/10.1109/ICDE55515.2023.00056

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. arXiv:2303.14524 [cs.IR] https://arxiv.org/abs/2303.14524

Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022a. Explainable Fairness in Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid Spain, 681–691. https://doi.org/10.1145/3477495.3531973

Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022b. Explainable Fairness in Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. ACM, 681–691. https://doi.org/10.1145/3477495.3531973

Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 196–204. https://doi.org/10.1145/3336191.3371824 arXiv:1911.08378 [cs, stat].

Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. A Re-visit of the Popularity Baseline in Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1749–1752. https://doi.org/10.1145/3397271.3401233 arXiv:2005.13829 [cs].

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* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 4717–4726. https://doi.org/10.1145/3589334.3648158

Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, Torino Italy, 1491–1494. https://doi.org/10.1145/3269206.3269264

Jie Li, Yongli Ren, Mark Sanderson, and Ke Deng. 2024. Explaining Recommendation Fairness from a User/Item Perspective. *ACM Trans. Inf. Syst.* 43, 1, Article 17 (Nov. 2024), 30 pages. https://doi.org/10.1145/3698877

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. LLaRA: Large Language-Recommendation Assistant. arXiv:2312.02445 [cs.IR] https://arxiv.org/abs/2312.02445

Luyang Lin, Lingzhi Wang, Jinsong Guo, and Kam-Fai Wong. 2024a. Investigating Bias in LLM-Based Bias Detection: Disparities between LLMs and Human Perception. arXiv:2403.14896 [cs.CY] https://arxiv.org/abs/2403.14896

Siyi Lin, Sheng Zhou, Jiawei Chen, Yan Feng, Qihao Shi, Chun Chen, Ying Li, and Can Wang. 2024b. ReCRec: Reasoning the Causes of Implicit Feedback for Debiased Recommendation. *ACM Trans. Inf. Syst.* 42, 6, Article 158 (Oct. 2024), 26 pages. https://doi.org/10.1145/3672275

Xu Liu, Tong Yu, Kaige Xie, Junda Wu, and Shuai Li. 2024. Interact with the explanations: Causal debiased explainable recommendation system. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 472–481.

Zhenghao Liu, Sen Mei, Chenyan Xiong, Xiaohua Li, Shi Yu, Zhiyuan Liu, Yu Gu, and Ge Yu. 2023. Text Matching Improves Sequential Recommendation by Reducing Popularity Biases. arXiv:2308.14029 [cs.IR] https://arxiv.org/abs/2308.14029

Weishen Pan, Sen Cui, Jiang Bian, Changshui Zhang, and Fei Wang. 2021. Explaining Algorithmic Fairness Through Fairness-Aware Causal Path Decomposition. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event Singapore, 1287–1297. https://doi.org/10.1145/3447548.3467258

Joey De Pauw, Koen Ruymbeek, and Bart Goethals. 2022. Modelling Users with Item Metadata for Explainable and Interactive Recommendation. arXiv:2207.00350 [cs.IR] https://arxiv.org/abs/2207.00350

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation Learning with Large Language Models for Recommendation. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*. ACM, 3464–3475. https://doi.org/10.1145/3589334.3645458

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. arXiv:1205.2618 [cs.IR] https://arxiv.org/abs/1205.2618

Ítallo Silva, Leandro Marinho, Alan Said, and Martijn C. Willemsen. 2024. Leveraging ChatGPT for Automated Human-centered Explanations in Recommender Systems. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (Greenville, SC, USA) *(IUI '24)*. Association for Computing Machinery, New York, NY, USA, 597–608. https://doi.org/10.1145/3640543.3645171

Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 837–847. https://doi.org/10.1145/3366423.3380164

Zuoli Tang, Zhaoxin Huan, Zihao Li, Shirui Hu, Xiaolu Zhang, Jun Zhou, Lixin Zou, and Chenliang Li. 2024a. TEXT CAN BE FAIR: Mitigating Popularity Bias with PLMs by Learning Relative Preference. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 2240–2249. https://doi.org/10.1145/3627673.3679581

Zuoli Tang, Zhaoxin Huan, Zihao Li, Shirui Hu, Xiaolu Zhang, Jun Zhou, Lixin Zou, and Chenliang Li. 2024b. TEXT CAN BE FAIR: Mitigating Popularity Bias with PLMs by Learning Relative Preference. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 2240–2249. https://doi.org/10.1145/3627673.3679581

Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. 2018. Attacks Meet Interpretability: Attribute-steered Detection of Adversarial Samples. http://arxiv.org/abs/1810.11580 arXiv:1810.11580 [cs, stat].

Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. ACM, 1717–1725. https://doi.org/10.1145/3447548.3467249

Xi Wang, Hossein A. Rahmani, Jiqun Liu, and Emine Yilmaz. 2023. Improving Conversational Recommendation Systems via Bias Analysis and Language-Model-Enhanced Data Augmentation. arXiv:2310.16738 [cs.CL] https://arxiv.org/abs/2310.16738

Yuyan Wang, Pan Li, and Minmin Chen. 2025. The Blessing of Reasoning: LLM-Based Contrastive Explanations in Black-Box Recommender Systems. arXiv:2502.16759 [cs.IR] https://arxiv.org/abs/2502.16759

Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. 2024. Can Small Language Models be Good Reasoners for Sequential Recommendation? arXiv:2403.04260 [cs.IR] https://arxiv.org/abs/2403.04260

Kathrin Wardatzky, Oana Inel, Luca Rossetto, and Abraham Bernstein. 2025. Whom do Explanations Serve? A Systematic Literature Survey of User Characteristics in Explainable Recommender Systems Evaluation. arXiv:2412.14193 [cs.HC] https://arxiv.org/abs/2412.14193

Jacek Wasilewski and Neil Hurley. 2016. Incorporating Diversity in a Learning to Rank Recommender System.. In *FLAIRS*. 572–578.

Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. ACM, 1791–1800. https://doi.org/10.1145/3447548.3467289

Chen Xu, Wenjie Wang, Yuxin Li, Liang Pang, Jun Xu, and Tat-Seng Chua. 2024. A Study of Implicit Ranking Unfairness in Large Language Models. arXiv:2311.07054 [cs.IR] https://arxiv.org/abs/2311.07054

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025a. Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models. arXiv:2501.09686 [cs.AI] https://arxiv.org/abs/2501.09686

Wujiang Xu, Yunxiao Shi, Zujie Liang, Xuying Ning, Kai Mei, Kun Wang, Xi Zhu, Min Xu, and Yongfeng Zhang. 2025b. iAgent: LLM Agent as a Shield between User and Recommender Systems. arXiv:2502.14662 [cs.CL] https://arxiv.org/abs/2502.14662

Guipeng Xv, Xinyi Liu, Chen Lin, Hui Li, Chenliang Li, and Zhenhua Huang. 2022. Lightweight Unbiased Multi-teacher Ensemble for Review-based Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 4620–4624. https://doi.org/10.1145/3511808.3557629

Emre Yalcin and Alper Bilge. 2021. Investigating and counteracting popularity bias in group recommendations. *Information Processing & Management* 58, 5 (2021), 102608.

Mengyuan Yang, Mengying Zhu, Yan Wang, Linxun Chen, Yilei Zhao, Xiuyuan Wang, Bing Han, Xiaolin Zheng, and Jianwei Yin. 2024. Fine-tuning large language model based explainable recommendation with explainable quality reward. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'24/IAAI'24/EAAI'24)*. AAAI Press, Article 1029, 10 pages. https://doi.org/10.1609/aaai.v38i8.28777

Wenzhuo Yang, Jia Li, Chenxi Li, Latrice Barnett, Markus Anderle, Simo Arajarvi, Harshavardhan Utharavalli, Caiming Xiong, and Steven HOI. 2021. On the Diversity and Explainability of Recommender Systems: A Practical Framework for Enterprise App Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 4302–4311. https://doi.org/10.1145/3459637.3481940

Xiaohan Yu, Li Zhang, and Chong Chen. 2024. Explainable CTR Prediction via LLM Reasoning. arXiv:2412.02588 [cs.IR] https://arxiv.org/abs/2412.02588

Jingsen Zhang, Xu Chen, Jiakai Tang, Weiqi Shao, Quanyu Dai, Zhenhua Dong, and Rui Zhang. 2023a. Recommendation with Causality enhanced Natural Language Explanations. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 876–886. https://doi.org/10.1145/3543507.3583260

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023c. AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems. arXiv:2310.09233 [cs.IR] https://arxiv.org/abs/2310.09233

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. arXiv:1904.09675 [cs.CL] https://arxiv.org/abs/1904.09675

Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101. https://doi.org/10.1561/1500000066

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 *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. ACM, 11–20. https://doi.org/10.1145/3404835.3462875

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023b. CoLLM: Integrating Collaborative Embeddings into Large Language Models for Recommendation. arXiv:2310.19488 [cs.IR]

Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Y. Lim. 2022. Exploiting Explanations for Model Inversion Attacks. arXiv:2104.12669 [cs.CV] https://arxiv.org/abs/2104.12669

Yuying Zhao, Yu Wang, Yunchao Liu, Xueqi Cheng, Charu C Aggarwal, and Tyler Derr. 2025. Fairness and diversity in recommender systems: a survey. *ACM Transactions on Intelligent Systems and Technology* 16, 1 (2025), 1–28.

Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity Bias in Dynamic Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event Singapore, 2439–2449. https://doi.org/10.1145/3447548.3467376

Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2022. Evolution of Popularity Bias: Empirical Study and Debiasing. http://arxiv.org/abs/2207.03372 arXiv:2207.03372 [cs].