PureBiasoMeter: Decoupling Popularity Bias from User Fairness in LLM-Based Recommender Systems

Shirin Tahmasebi*, Muhammad Hamad*, Amir H. Payberah, and Mihhail Matskin

KTH University of Technology, Stockholm, Sweden
{shirint, mhama, payberah, misha}@kth.se

Abstract. Large Language Models (LLMs) have transformed Recommendation Systems (RecSys) by enabling more user-aware interactions. However, this shift raises new challenges in evaluating fairness, particularly due to confounding systemic biases such as *popularity bias*. Conventional fairness assessments often confuse disparities in user treatment with systemic biases toward popular items, resulting in misleading conclusions. In this paper, we introduce **PureBiasoMeter**, a diagnostic framework that decouples popularity bias from user-level fairness in LLM-based RecSys. Our approach is based on the hypothesis that accurate fairness evaluation requires first mitigating popularity bias and then remeasuring fairness metrics. We generate a comprehensive set of 72 prompts using different user profiling strategies, demographic variants, and bias mitigation instructions. By applying these prompts in a black-box LLM setting, we evaluate fairness sensitivity, recommendation quality, and bias levels across multiple dimensions. Our results demonstrate that removing popularity bias substantially alters fairness measurements and reveals underlying disparities that were previously unknown. **PureBiasoMeter** thus provides a more reliable basis for fairness analysis and contributes a practical tool for disentangling intertwined sources of bias in modern RecSys.

Keywords: Recommendation Systems · Large Language Models · User Fairness · Popularity Bias.

1 Introduction

Large Language Models (LLMs) have significantly advanced the capabilities of Recommendation Systems (RecSys), enabling sophisticated, user-centered interactions. However, these developments also raise serious concerns about *bias* and *fairness*, which can undermine both the reliability and societal impact of such systems. In fairness studies, it is essential to distinguish between these two concepts, which, while related, have distinct meanings and implications [2]. *Fairness*

[*] The first and second authors contributed equally.

focuses on the equitable treatment of users and alignment with societal values, whereas *bias* refers to systemic deviations from a target distribution, such as favoring overly popular items, a phenomenon known as *popularity bias* [1]. Among various biases in LLM-based RecSys, popularity bias is particularly problematic, as it reduces content diversity and, more critically, interferes with fairness evaluations by masking user-level disparities [5].

Recent works, such as [1, 4, 10], have addressed popularity bias and user unfairness, but usually treat them separately. Some studies attempt to reduce popularity bias using prompt-based techniques [8], while others test user fairness by injecting sensitive attributes into prompts and observing recommendation differences [12]. However, such fairness evaluations often overlook the influence of underlying biases, such as popularity. This makes it difficult to determine whether unfair treatment stems from user attributes or from the model's built-in preference for popular items. Hence, fairness assessments risk overgeneralization and misdiagnosis, resulting in misleading conclusions about model fairness.

This reveals a critical research gap: fairness evaluations in LLM-based RecSys often conflate the effects of different biases with actual unfair treatment of users. In particular, when popularity bias is not properly accounted for, it can distort fairness assessments by making some user groups appear to receive better or worse recommendations, not because of who they are, but because of how the model favors popular items. This makes it difficult to determine whether the system is truly treating users unfairly.

While popularity bias is intrinsic to many RecSys models, its confounding effect on fairness evaluation can be amplified in LLM-based RecSys. Their tendency to overproduce popular titles can artificially equalize outputs across demographic groups, leading to inflated fairness scores. We therefore hypothesize that mitigating the influence of popularity bias is necessary to accurately evaluate user fairness. By isolating this effect, we can gain a clearer understanding of how the model treats users. However, our goal is not to eliminate such biases entirely, but to measure and quantify their influence on fairness diagnostics.

To address this, we introduce **PureBiasoMeter**, a diagnostic framework designed to decouple popularity bias from user-level fairness assessments in LLM-based RecSys. Rather than proposing a new fairness metric or mitigation algorithm, **PureBiasoMeter** provides a structured approach to *purifying* fairness evaluations by factoring out popularity effects. To do so, we construct a factorial evaluation using 72 systematically varied prompts that differ in user profiling strategy, demographic sensitivity, and popularity bias mitigation instructions. We then assess the quality of recommendations, fairness sensitivity, and popularity bias using metrics such as Hit Ratio (HR), Jaccard Similarity (JS), and Log Popularity Difference (LPD).

Our findings demonstrate that: (1) Popularity bias can significantly mask disparities in fairness evaluations; (2) Reducing popularity bias results in large shifts in fairness metrics, revealing hidden unfairness; and (3) Certain combinations, especially `polarized` profiling and `temporal-diverse` instructions, can effectively reduce popularity bias without sacrificing personalization quality.

2 Problem Definition

To define the problem, we begin by clarifying the distinction between *fairness* and *bias* in the context of LLM-based RecSys [2]. *Fairness* refers to the principle that users should be treated equitably, especially with respect to sensitive attributes such as gender [2, 12]. In RecSys, fairness is commonly evaluated by generating recommendations for user profiles that differ only in demographic attributes and comparing the outputs [3, 12]. If the results vary significantly, this may indicate unfair treatment. *Bias* refers to systemic deviations from an objective target distribution [1, 2]. A particularly critical bias in LLM-based RecSys is *popularity bias*, the tendency to recommend widely popular items regardless of users' actual preferences [5, 8].

The key issue is that popularity bias can distort fairness evaluations. When a model recommends the same set of popular items to all users, it may seem to treat them fairly because the outputs appear similar. However, this similarity might reflect the system's popularity bias rather than its fair treatment. In such cases, fairness metrics become unreliable because they reflect a combination of popularity bias and user demographics sensitivity.

For example, consider two users with highly distinct viewing histories but the same age and similar engagement levels. If the model still recommends the same trending titles to both, it gives the illusion of fairness because the outputs overlap substantially. Yet this overlap stems from popularity bias; the model defaults to recommending what is most popular, not what best matches each user's preferences. Once popularity bias is mitigated, the recommendations begin to differ, reflecting users' actual interests. This difference does not indicate new unfairness; it simply shows that the earlier similarity came from popularity bias, not from truly fair treatment.

In summary, current fairness evaluations in LLM-based RecSys often mix up popularity bias with user unfairness. To make reliable conclusions, we need to decouple these two disparities.

3 PureBiasoMeter

PureBiasoMeter is a diagnostic framework designed to estimate how much of the observed unfairness in LLM-based RecSys is attributable to popularity bias. Our core idea is simple: instead of directly measuring user fairness in the presence of confounding factors, we first reduce popularity bias and then remeasure fairness. Comparing fairness metrics before and after debiasing enables us to assess whether earlier measurements were distorted by popularity bias; larger differences indicate greater distortion.

To this end, we design an LLM-based RecSys that generates recommendations through carefully structured prompts, enabling controlled experimentation without requiring model fine-tuning or architectural changes. This setup enables us to isolate our analysis from confounding implementation factors and focus specifically on how prompt structure and bias interventions affect fairness evaluation. We decompose each prompt into three key components: *(1) User Profiling*, *(2) User Demographics*, and *(3) Bias Mitigation Instructions*.

User Profiling: This component summarizes the user's interaction history in a compact form suitable for prompting. Since real-world user histories can be long, it is not feasible to include all interactions in a single prompt. Instead, we extract a concise summary that captures the user's preferences. The key question is: *which interactions should be included to best reflect the user's interests?* To address this, we adopt several strategies for constructing the summary:

- `top-rated`: Select the five items with the highest user ratings, representing the user's strongest preferences.
- `most-recent`: Select the five most recently interacted items, reflecting the user's current interests.
- `polarized`: Select a mix of the user's highest- and lowest-rated items. This strategy provides a more nuanced understanding of both positive and negative preferences, potentially enabling the model to differentiate between liked and disliked content.

User Demographics: This component injects demographic attributes into the prompt to enable fairness evaluation. We design multiple variants to support both single-attribute and intersectional fairness analysis. These variants allow us to control the presence and combination of user-sensitive attributes in the prompt:

- `neutral`: No demographic information is included. The prompt reflects a generic user without any identity markers.
- `gender-age`: Includes both gender and age in the description. (e.g., "25-year-old male").
- `occupation-only`: Includes only the user's profession (e.g., "The user works as a software engineer").
- `all`: Combines all available demographic details (e.g., "The user is a 25-year-old male who works as a software engineer.")

Bias Mitigation Instructions: This component is used to influence the model's recommendation behavior with respect to item popularity. We design six variations, each targeting a different aspect of popularity bias:

- `baseline`: In this case, there is no mention of popularity. The prompt simply requests recommendations based on the user profile, which reflects the LLM's default behavior.
- `niche-genre`: Instructs the model to focus on lesser-known or narrowly defined genres from the genres the user has interacted with in the past.
- `exclude-popular`: Explicitly instructs the model to avoid recommending very popular or blockbuster items.

- `lesser-known`: Directs the model to recommend items that are less mainstream in terms of production origin or cultural exposure, such as independent or non-Western content. This differs from `niche-genre` in that it focuses on content provenance and production companies rather than genre.
- `temporal-diverse`: Encourages recommendations that span across different time periods, promoting temporal diversity and reducing recency bias.
- `obscure-theme`: Instructs the model to suggest content with unconventional and experimental themes, irrespective of their popularity level [1].

By combining various strategies for each component, we generate a total of 72 distinct prompts (3 user profiling × 4 user demographics × 6 bias mitigation instructions). These permutations enable us to comprehensively investigate how different prompt constructions impact recommendation behavior and fairness metrics. After generating recommendations using the above prompts, we apply lightweight postprocessing to clean the outputs.

Evaluation Metrics. We categorize our evaluation metrics into three groups: *(1) Recommendation Quality*, *(2) Popularity Bias*, and *(3) Fairness Sensitivity.* Each group captures a distinct aspect of model behavior under prompt variation.

Recommendation Quality: We use Hit Ratio at rank k (HR@k) to evaluate recommendation utility. It indicates whether at least one relevant item appears in the top-k results. We report the average HR across all prompts; higher values indicate better performance.

Popularity Bias: To measure the extent to which the model favors popular content, we use the Log Popularity Difference (LPD) metric from [8]. This is computed as:

$$\mathrm{LPD} = \log\left(\frac{1}{|R|}\sum_{i\in R}\mathrm{pop}(i)\right) - \log\left(\frac{1}{|H|}\sum_{j\in H}\mathrm{pop}(j)\right),$$

where: R is the set of recommended items; H is the set of items previously rated or interacted by the user; and $\mathrm{pop}(i)$ is the frequency (e.g., rating count) of item i in the training dataset.

This log-based transformation reduces the impact of extremely popular outliers and ensures that the values are *centered* around *zero*. A *positive* LPD indicates that the recommendations are skewed toward more popular items than the user typically consumes, suggesting potential popularity bias. Conversely, a *negative LPD* implies that the system recommends less popular items compared to the user's historical preferences.

[1] Standard genres like comedy or drama offer broad labels but often miss the nuanced themes users care about, such as emotionally introspective road trip stories with humor. Obscure themes capture such specific narrative preferences beyond genre categories.

Fairness Sensitivity: To assess fairness, we measure how sensitive recommendations are to changes in sensitive attributes. Specifically, as in [12], we compute the Jaccard Similarity (JS) between the recommendation set for a sensitive user profile and that of the corresponding neutral profile:

$$\text{JS}(s, \texttt{neutral}) = \frac{|R_s \cap R_n|}{|R_s \cup R_n|},$$

where R_s is the set of items recommended to the sensitive variant and R_n to the neutral one. A low similarity score indicates that changing demographic attributes substantially alters recommendations, which may reflect unfair treatment.

In general, this multi-metric evaluation enables us to analyze how prompt strategies impact recommendation quality, bias, and fairness in a disentangled and interpretable manner.

4 Experiment Design

We have designed several experiments to evaluate the core hypothesis behind **PureBiasoMeter**: that fairness evaluations in LLM-based RecSys can be misleading if popularity bias is not first reduced. Specifically, we aim to determine whether existing fairness assessments mistakenly interpret popularity bias as evidence of unfair user treatment, and whether decoupling popularity bias from user unfairness is essential for accurate fairness assessment. Now, in what follows, we describe these experiments:

- *Fairness Shift from Popularity Bias Reduction*: Here, we want to answer this question: *Does reducing popularity bias significantly alter fairness evaluations?* To do so, we analyze JS to determine whether fairness measurements (based on JS between sensitive and neutral prompts) significantly change after mitigating popularity bias. A large change in JS would suggest that previous fairness measurements were distorted by popularity effects.

- *Prompt Design Sensitivity*: The main question evaluated in this experiment is: *Which parts of the prompt most strongly affect fairness, personalization, and popularity bias?* To answer this, we explore how variations in prompt structure, such as user profiling, user demographics, and bias mitigation instructions, affect recommendation outcomes. For each configuration, we compute JS and HR to assess how different prompt components influence recommendation quality, fairness, and popularity bias.

- *Bias–Personalization Trade-off*: The question is: *Is it possible to improve bias without degrading personalization quality?* We analyze the relationship between raw HR and JS values across prompt variants to examine whether improved bias comes at the cost of reduced utility.

- *Bias Reduction Effectiveness*: Finally, the question is: *Does our intervention successfully reduce popularity bias in the output?* To answer it, we compute LPD to verify if our popularity bias mitigation is actually effective.

Together, these experiments evaluate whether **PureBiasoMeter** enables more decoupled and accurate fairness assessment compared to conventional evaluations that ignore popularity bias.

Implementation Details. To ensure comparability with prior work [12], we adopt a similar evaluation setup using a closed-source LLM, `GPT-4.1-nano` [2]. We interact with the model solely via prompting, without any fine-tuning, internal access, or architectural modification. This design allows us to directly assess fairness and popularity bias in a controlled, black-box setting and compare with other fairness evaluation methods.

Each prompt configuration is used to generate ten recommendations, using fixed decoding parameters (e.g., temperature and top-k sampling) for consistency. Popularity scores are computed from rating frequencies in the MovieLens dataset [6]. All reported results are averaged over all users in the dataset. Code and evaluation templates are publicly available [3].

5 Results and Analysis

We experimented with all combinations of profiling strategies, user demographics, and popularity bias interventions (see Appendix A for the full results table). Due to the table's size, we organize this section around the key experimental questions from Section 4. Each subsection presents the most relevant results and insights for a specific question, utilizing a modular structure to enhance clarity and readability.

5.1 Fairness Shift from Popularity Bias Reduction

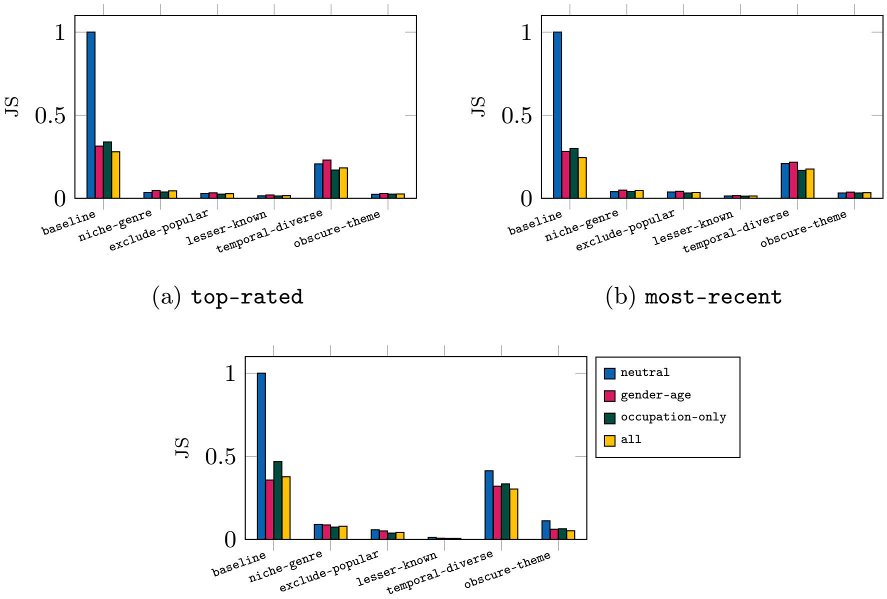
Question: Does reducing popularity bias significantly alter fairness evaluations?

Figures 1a, 1b, and 1c illustrate how JS between recommendations generated with neutral prompts and those with fairness-sensitive prompts varies across different popularity bias mitigation strategies and profiling settings. The JS for the `neutral` group under the `baseline` condition is always 1. This reflects a self-comparison between the neutral prompt and itself, serving as a natural reference point rather than a meaningful fairness signal.

At first glance, the high JS in the baseline condition might suggest that the model treats all users fairly, that is, recommendations for fairness-sensitive prompts closely resemble those for the neutral prompt. However, these high similarities are largely driven by the dominance of popular content: because highly popular items are recommended to nearly everyone, differences across demographic prompts are suppressed, and the outputs appear *superficially* fair.

[2] `https://platform.openai.com/docs/models/gpt-4.1-nano`

[3] `https://github.com/Hamad-Security/LLM-Bias-Fairness-Assessment`



(a) `top-rated`

(b) `most-recent`

(c) `polarized`

Fig. 1: Fairness Sensitivity (JS) Across Popularity Mitigation Strategies. The JS for the `neutral` group under the `baseline` condition is always 1, which reflects a self-comparison between the neutral prompt and itself.

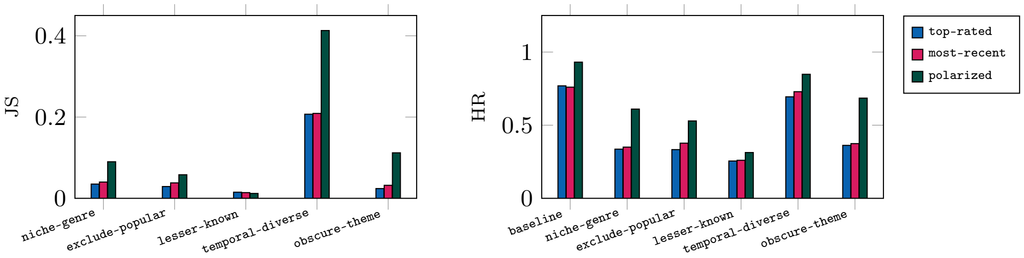
Once we apply popularity bias mitigation (e.g., `niche-genre`, `exclude-popular`, or `lesser-known`), the JS drops sharply and consistently across all profiling strategies. This drop reveals that much of the apparent fairness observed in the baseline setting was actually the result of popularity bias. When this confounding factor is removed, meaningful variation between demographic groups begins to emerge.

The contrast between the high baseline JS and the much lower post-mitigation scores demonstrates that popularity can mask actual disparities in user treatment. Without controlling for popularity, fairness evaluations risk confusing popularity bias with user unfairness. These findings confirm our hypothesis: accurate fairness diagnostics in LLM-based RecSys require the decoupling of popularity bias from demographic unfairness.

5.2 Prompt Design Sensitivity

Question: Which parts of the prompt most strongly affect fairness, personalization, and popularity bias?

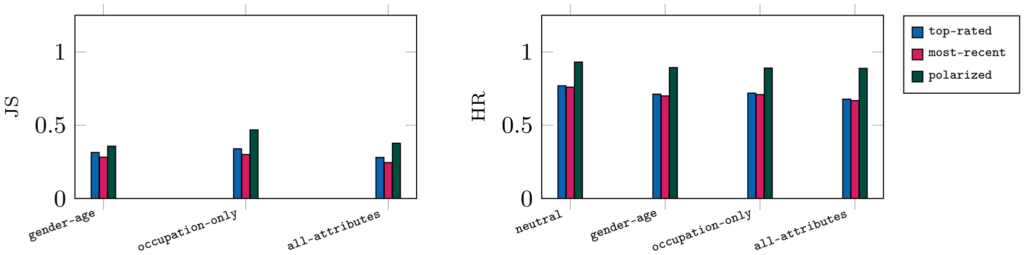
Figures 2 and 3 investigate how different components of the prompt, namely, user profiling, user demographic description, and bias mitigation instruction, can individually affect recommendation behavior in terms of fairness (JS) and quality (HR).



(a) JS

(b) HR

Fig. 2: JS and HR Across Instruction Variants (Fairness = `neutral`)



(a) JS

(b) HR

Fig. 3: JS and HR Across Fairness Variants (Popularity Bias = `baseline`)

In Figures 2a and 2b, we fix fairness to the `neutral` setting and vary the bias mitigation instruction provided to the model. Among the three profiling strategies, `polarized` consistently yields the highest JS scores, particularly for some of the popularity bias strategies, such as `temporal-diverse` and `obscure-theme`. This suggests that the user profiling approach and bias mitigation instruction component play a strong role in controlling personalization and fairness.

Turning to Figure 3a, we fix the popularity bias strategy (`baseline`) and vary the fairness objective across different demographic attributes. Here again, `polarized` prompts stand out with higher JS scores across all fairness variants. Importantly, this increase in fairness-aware diversity does not lead to a reduction in quality; HR remains high across all settings, reinforcing that fairness improvements can be achieved without sacrificing personalization.

The main takeaway of this experiment is that the user profiling strategy embedded in the prompt plays the most decisive role in shaping fairness, personalization, and sensitivity to popularity bias. While user demographic and popularity bias instructions can also affect the recommendation behavior, their effectiveness depends heavily on the underlying profiling strategy.

5.3 Bias–Personalization Trade-off

Question: Is it possible to improve bias without degrading personalization quality?

To evaluate the trade-off between fairness and personalization in LLM-based RecSys, we analyze the impact of popularity bias mitigation strategies on both JS and HR. In this experiment, we fix the fairness prompt to `neutral`, ensuring that no user demographic attributes are injected. This allows us to isolate the effect of popularity debiasing alone on fairness evaluation and recommendation performance.

Figure 2 summarizes the results across different profiling strategies. In Figure 2a, we observe how JS varies with different bias instructions. In the `baseline` setting, JS is uniformly 1 across all profiles, which is due to self-comparison; therefore, we excluded it from the figure. For other popularity settings, once they are applied, JS drops substantially. For instance, strategies like `lesser-known` and `exclude-popular` yield the lowest JS values, indicating that the model begins to differentiate more between `baseline` and other popularity bias mitigation prompts.

However, this clarification of popularity bias and its unmasking comes with a cost. Figure 2b shows that HR also declines under some of the debiasing strategies. The `lesser-known` and `exclude-popular` instructions result in the largest drop in HR, reflecting a degradation in recommendation quality when popular content is suppressed. On the other hand, some strategies, such as `temporal-diverse`, strike a better balance: both JS and HR are closer to the ideal case, especially under the `polarized` profile.

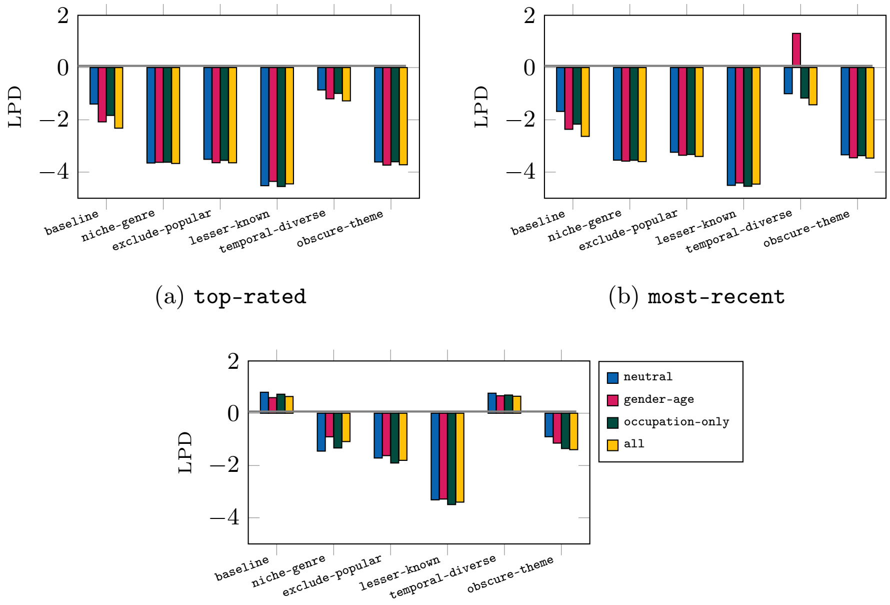
5.4 Bias Reduction Effectiveness

Question: Does our intervention successfully reduce popularity bias?

To evaluate if our interventions successfully mitigate popularity bias, we analyze LPD across multiple prompt configurations. LPD is designed to be zero-centered: an ideal RecSys that matches a user's historical preferences in terms of item popularity should have an LPD close to 0.

Figure 4 presents LPD values under various popularity bias mitigation strategies across three profiling methods, `top-rated`, `most-recent`, and `polarized`, and four fairness prompt variants. We observe that in the `baseline` condition (i.e., no popularity mitigation), LPD values have high absolute values and far from zero. This confirms the presence of popularity bias in the model's default recommendation behavior, as it tends to favor globally popular items over content aligned with a user's own interaction history.

Once we apply debiasing instructions, the LPD values change significantly. Among the strategies, `temporal-diverse` consistently produces LPD values that are both lower and more stable across fairness prompts, indicating that it is the most robust in reducing popularity bias without overcorrecting. Interestingly, we also observe that the effectiveness of debiasing strategies varies depending on the profiling method. The `polarized` profile tends to result in the best LPDs



(a) `top-rated`

(b) `most-recent`

(c) `polarized`

Fig. 4: LPD Variation Across Popularity Mitigation Strategies

across most configurations. This suggests that it captures a more nuanced representation of user preferences, making it easier for the model to deviate from default popularity trends.

These findings confirm that our intervention successfully reduces the confounding effect of popularity bias in the recommendation process.

6 Related Work

We briefly review related work on fairness and popularity bias in RecSys.

Fairness in RecSys. Prior work has proposed frameworks for modeling and evaluating fairness using individual- and group-level metrics [4], and highlighted how demographic and popularity biases distort recommendation outcomes [5]. Counterfactual fairness offers a way to isolate the effect of sensitive attributes [7], while other studies examine fairness–utility trade-offs [9] [10]. In LLM-based RecSys, fairness has been tested via prompt sensitivity to demographics [12], but often without accounting for confounding popularity bias.

Popularity Bias in RecSys. Popularity bias is a well-known problem in RecSys. The study in [1] was among the first to formalize the concept of popularity bias in ranking algorithms, showing its negative effect on diversity and minority exposure. The authors in [11] further analyzed how item popularity distorts recommendation accuracy and proposed evaluation techniques that balance personalization with fairness. More recently, researchers in [8] studied popularity bias in LLM-based RecSys, highlighting how LLMs exhibit strong preferences for mainstream content.

7 Conclusion

We introduced **PureBiasoMeter**, a diagnostic framework for decoupling popularity bias from user unfairness in LLM-based RecSys. Our results show that fairness metrics are often distorted by popularity bias, which can mask true disparities. By mitigating popularity effects before evaluation, **PureBiasoMeter** reveals hidden unfairness.

Using a factorial prompt design, we analyzed how profiling strategies, demographics, and bias mitigation instructions affect recommendations. Notably, `polarized` profiling with `temporal-diverse` instructions improved fairness sensitivity without harming recommendation quality. **PureBiasoMeter** enables more reliable fairness diagnostics and provides a foundation for future interventions. It can also be extended to disentangle other systemic biases, such as position or recency bias.

Appendix A: Full Results

This appendix contains the complete set of results from our factorial evaluation across all prompt configurations in Table 1. To enhance interpretability, we apply a heatmap-style color scheme to each column of the results table. Within each metric, darker shades indicate better performance, while lighter ones indicate worse outcomes.

Table 1: Full Evaluation Results Across Prompt Configurations

		`top-rated`			`most-recent`			`polarized`		
Fairness	**Bias**	**HR**	**JS**	**LPD**	**HR**	**JS**	**LPD**	**HR**	**JS**	**LPD**
`neutral`	`baseline`	0.769	–	-1.395	0.760	–	-1.678	0.931	–	0.803
	`niche-genre`	0.336	0.035	-3.651	0.350	0.040	-3.542	0.610	0.090	-1.449
	`exclude-popular`	0.333	0.029	-3.510	0.377	0.038	-3.240	0.529	0.058	-1.713
	`lesser-known`	0.255	0.015	-4.520	0.260	0.014	-4.507	0.313	0.012	-3.314
	`temporal-diverse`	0.694	0.207	-0.856	0.729	0.209	-1.000	0.848	0.413	0.767
	`obscure-theme`	0.362	0.024	-3.611	0.374	0.032	-3.338	0.685	0.112	-0.899
`gender-age`	`baseline`	0.712	0.314	-2.077	0.700	0.282	-2.362	0.893	0.357	0.597
	`niche-genre`	0.358	0.047	-3.625	0.369	0.049	-3.579	0.680	0.087	-0.903
	`exclude-popular`	0.344	0.033	-3.641	0.373	0.042	-3.355	0.585	0.051	-1.622
	`lesser-known`	0.284	0.020	-4.355	0.278	0.016	-4.415	0.316	0.007	-3.283
	`temporal-diverse`	0.737	0.230	-1.195	0.773	0.217	-1.307	0.876	0.320	0.669
	`obscure-theme`	0.364	0.029	-3.732	0.379	0.037	-3.452	0.663	0.061	-1.143
`occupation-only`	`baseline`	0.719	0.339	-1.831	0.709	0.300	-2.164	0.890	0.468	0.728
	`niche-genre`	0.349	0.038	-3.621	0.352	0.041	-3.547	0.626	0.074	-1.329
	`exclude-popular`	0.327	0.025	-3.549	0.359	0.032	-3.326	0.511	0.038	-1.904
	`lesser-known`	0.242	0.014	-4.551	0.247	0.013	-4.543	0.255	0.006	-3.495
	`temporal-diverse`	0.682	0.170	-0.989	0.720	0.168	-1.165	0.852	0.334	0.696
	`obscure-theme`	0.363	0.025	-3.605	0.382	0.032	-3.371	0.623	0.064	-1.352
`all`	`baseline`	0.678	0.280	-2.318	0.668	0.245	-2.633	0.888	0.377	0.640
	`niche-genre`	0.362	0.045	-3.670	0.369	0.047	-3.599	0.656	0.079	-1.084
	`exclude-popular`	0.326	0.028	-3.643	0.366	0.035	-3.401	0.547	0.042	-1.803
	`lesser-known`	0.266	0.016	-4.450	0.270	0.014	-4.458	0.268	0.006	-3.401
	`temporal-diverse`	0.699	0.183	-1.274	0.736	0.176	-1.424	0.869	0.303	0.650
	`obscure-theme`	0.367	0.026	-3.719	0.382	0.034	-3.463	0.623	0.052	-1.394

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Popularity bias in ranking and recommendation. In: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems. pp. 1–6 (2019)
2. Dai, S., Xu, C., Xu, S., Pang, L., Dong, Z., Xu, J.: Bias and unfairness in information retrieval systems: New challenges in the llm era. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 6437–6447 (2024)
3. Deldjoo, Y., Di Noia, T.: Cfairllm: Consumer fairness evaluation in large-language model recommender system. ACM Trans. Intell. Syst. Technol. (Mar 2025). https://doi.org/10.1145/3725853, https://doi.org/10.1145/3725853
4. Ekstrand, M.D., Burke, R., Diaz, F., Ekstrand, J.A., Pera, S.: Fairrec: Models, metrics, and methodology for fair recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1567–1576 (2022)
5. Ekstrand, M.D., Tian, M., Azpiazu, I., Ekstrand, J.A., Anuyah, O., McNeill, D., Pera, S.: All the cool kids, how do you recommend?: Popularity and demographic biases in recommender evaluation and effectiveness. Conference on Fairness, Accountability and Transparency (FAT*) (2018)
6. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
7. Kusner, M.J., Loftus, J., Russell, C., Silva, R.: Counterfactual fairness. Advances in Neural Information Processing Systems **30** (2017)
8. Lichtenberg, J.M., Buchholz, A., Schwöbel, P.: Large language models as recommender systems: A study of popularity bias. arXiv preprint arXiv:2406.01285 (2024)
9. Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., Diaz, F.: Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness and satisfaction in recommendation systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 2243–2251 (2018)
10. Singh, A., Joachims, T.: Fairness of exposure in rankings. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2219–2228 (2018)
11. Steck, H.: Item popularity and recommendation accuracy. In: Proceedings of the fifth ACM conference on Recommender systems. pp. 125–132. ACM (2011)
12. Zhang, J., Bao, K., Zhang, Y., Wang, W., Feng, F., He, X.: Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. pp. 993–999 (2023)