



Delft University of Technology

Document Version

Final published version

Citation (APA)

Hasami, A., & Mansoury, M. (2025). Mitigating Popularity Bias in Counterfactual Explanations using Large Language Models. In *RecSys2025 - Proceedings of the 19th ACM Conference on Recommender Systems* (pp. 1234-1239). Association for Computing Machinery (ACM). <https://doi.org/10.1145/3705328.3759330>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership. Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

**Green Open Access added to [TU Delft Institutional Repository](#)
as part of the Taverne amendment.**

More information about this copyright law amendment
can be found at <https://www.openaccess.nl>.

Otherwise as indicated in the copyright section:
the publisher is the copyright holder of this work and the
author uses the Dutch legislation to make this work public.



Mitigating Popularity Bias in Counterfactual Explanations using Large Language Models

Arjan Hasami
Delft University of Technology
Delft, Netherlands
a.hasami@student.tudelft.nl

Masoud Mansoury
Delft University of Technology
Delft, Netherlands
m.mansoury@tudelft.nl

Abstract

Counterfactual explanations (CFEs) offer a tangible and actionable way to explain recommendations by showing users a "what-if" scenario that demonstrates how small changes in their history would alter the system's output. However, existing CFE methods are susceptible to bias, generating explanations that might misalign with the user's actual preferences. In this paper, we propose a pre-processing step that leverages large language models to filter out-of-character history items before generating an explanation. In experiments on two public datasets, we focus on popularity bias and apply our approach to ACCENT, a neural CFE framework. We find that it creates counterfactuals that are more closely aligned with each user's popularity preferences than ACCENT alone.

CCS Concepts

• **Information systems** → Recommender systems; Presentation of retrieval results.

Keywords

recommender systems, counterfactual explanations, large language models, popularity bias

ACM Reference Format:

Arjan Hasami and Masoud Mansoury. 2025. Mitigating Popularity Bias in Counterfactual Explanations using Large Language Models. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25)*, September 22–26, 2025, Prague, Czech Republic. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3705328.3759330>

1 Introduction

Recommender systems (RS) have become vital for guiding users to relevant content across various domains [7, 28]. As these systems become increasingly complex, leveraging deep learning and vast amounts of data, the decision-making process becomes more opaque, raising concerns about transparency and trust [8]. This has resulted in the emergence of Explainable RS as an area of research. Various explanation approaches have been explored, but ensuring that explanations are faithful to the model's logic and meaningful to the user remains challenging [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

RecSys '25, Prague, Czech Republic

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1364-4/25/09

<https://doi.org/10.1145/3705328.3759330>

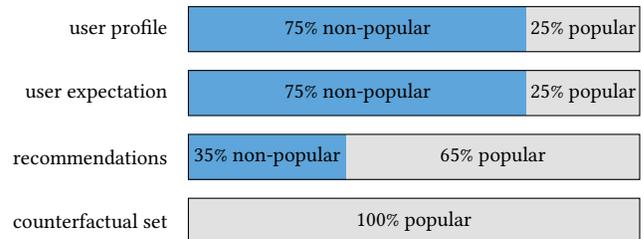


Figure 1: A user with both popular and non-popular items in their history may expect a similar balance in recommendations and counterfactual explanations. However, this alignment often breaks due to popularity bias. The effect can be even stronger in counterfactual sets, which are typically small and thus more sensitive to such skew.

Counterfactual explanations (CFEs) [9, 33] have emerged as an appealing solution to the tangibility problem in explainable RS. A CFE identifies a small set of the user's past interactions such that if those interactions were removed, the system's top recommendation would change. These personalized and actionable explanations allow users to see how their tastes drive the outcomes. Indeed, user studies have found that people appreciate detailed CFEs, favoring them over generic transparency statements [29].

A well-known issue with CFEs is the Rashomon effect [23]: multiple distinct explanations can be valid for the same recommendation, each highlighting a different set of changes that might contradict each other. While this diversity of possible explanations can be seen as a strength, it also raises the question of which explanation should be shown to the user. Prior work tackles this in several, non-mutually exclusive ways, such as keeping only the nearest edit [10, 30, 32] or returning a diverse set of multiple counterfactuals [3, 24]. Instead, we argue that the selected CFE should align with the user's expectations and interests. Highlighting an obscure or marginally relevant item may satisfy formal criteria, yet appear unintuitive or misleading to the user. Previous research also stresses that explanations should be tailored to the user and context, as different users value different criteria [5, 31].

The need for alignment becomes even more relevant when considering that explanation methods may reflect biases in the underlying recommender model. Because many CFE frameworks operate on internal signals such as gradients or influence measures, systemic biases can surface in the generated explanations. Figure 1 demonstrates how this might look with popularity bias [1, 15, 20, 21], where users mainly interested in niche items may fail to resonate with explanations based on popular items.

In this paper, we explore large language models (LLMs) as a tool to improve the user alignment of CFEs. LLMs possess a vast capacity to understand context, semantics, and nuances from textual data. Recently, LLMs have been leveraged in the recommendation task to turn a list of consumed items supplemented with descriptions into a coherent textual profile that captures themes a user consistently enjoys [27]. We embed that profile, greedily remove history items whose absence changes it the most, and run ACCENT, a CFE framework for neural recommenders [32], on this filtered history.

By eliminating out-of-character interactions before the explanation step, we cut down the space of admissible CFEs and guide ACCENT toward explanations more closely resembling users' preferences. As this augmentation relies on a general "does this suit the user?" test rather than on attribute-specific rules, it should, in principle, alleviate many kinds of mismatch. In this paper, we evaluate the approach solely on one well-known dimension, popularity bias. We present preliminary findings and show that it reduces mismatches between user popularity preferences and the popularity of items appearing in CFEs. Our contributions are threefold:

- We propose two metrics for detecting popularity bias in counterfactual sets and use them to empirically confirm that ACCENT's explanations can be skewed.
- We propose LLM-augmented counterfactual generation, a plug-in pre-processing step that can be paired with any history-based CFE method.
- We show that explanations generated on the filtered history more closely resemble the user's popularity preference on two public datasets.

2 Background

Counterfactual Explanations in RS. Counterfactual explanations in recommender systems take the following form:

You were recommended 'John Wick' because you previously liked 'Taken' and 'The Equalizer'. Otherwise, you would have been recommended 'Mean Girls'.

By identifying a change in a user's history that flips the system's top-1 recommendation, CFEs aim to provide personalized, actionable feedback. Prior CFE frameworks have taken various approaches, such as optimization-based frameworks that solve a constrained minimization problem [19, 30] or graph-centric approaches [10, 22]. We adopt ACCENT [32], a framework based on influence analysis, as our primary method for generating CFEs.

ACCENT in a nutshell. ACCENT [32] adapts influence functions [4, 16]—a technique from robust statistics [11]—to produce CFEs for neural recommenders. For a given user u , let r be the system's current top recommendation and r^* an alternative item (taken from the original top- k recommendations). ACCENT seeks the smallest subset E of the user's interaction history H_u whose removal would make the model prefer r^* over r . Formally,

$$\text{ACCENT}(H_u, \hat{y}, r, r^*) = \arg \min_{E \subseteq H_u} \{ |E| \mid \hat{y}_{u,r}^{-E} - \hat{y}_{u,r^*}^{-E} < 0 \} \quad (1)$$

where $\hat{y}_{u,r}^{-E}$ is the predicted score for user u and item r after virtually removing the interactions in counterfactual set E . ACCENT uses Influence functions to approximate these scores efficiently [4], avoiding retraining during candidate and subset evaluation.

3 Problem Formulation

Popularity bias in ACCENT. In recommender data, popular items converge the fastest [25]. As there are many more interactions, there are more training iterations and updates, giving them disproportionate weight during learning. ACCENT is based on influence functions, which tell us which training points are most used for a recommendation. However, due to the disproportionate weight of popular items due to popularity bias, these are more influential in most predictions. These popular items thus usually have a high influence score, making it likely that ACCENT picks them.

Towards bias-aware CFEs with LLMs. LLMs can encode high-level preferences, such as tone, themes, or narrative style, by reading free-text descriptions of consumed items. Recent work shows that LLM-based recommenders exhibit lower popularity bias than traditional baselines, and that careful prompting can reduce bias further [18]. We therefore investigate whether LLMs can act as a pre-processing step, filtering a user's history so that existing CFE frameworks such as ACCENT can start from a less biased candidate pool, leading to an explanation that aligns more with user interest.

4 LLM-Augmented ACCENT

User histories mix long-term tastes with one-off, context-driven interactions (e.g., watching a blockbuster with friends). These out-of-character items can skew CFE frameworks, such as ACCENT's influence scores. To reduce this step, we first distill the user's core profile by removing inconsistent items, shrinking the search space, and dampening the popularity signal before the CFE step.

We leverage LLMs to generate natural language profiles of user preferences based on their full interaction history and item descriptions. The prompt asks for a concise (<300 words) summary focused on recurring themes, tones, or character types—avoiding broad genres or direct item mentions. We then embed the resulting text into a high-dimensional vector. An example prompt and LLM response is as follows:

Example LLM Query and Response (Movie Domain)

Query: Your task is to analyze a list of movies a user has interacted with and describe the profile and the preferences of the user in less than 300 words. Try to not be too broad (e.g. mention too many general categories such as action or comedy). Do not mention specific movie titles.

The user has watched the following movies:
Die Hard (1988) - Action, Thriller - NYPD cop John...
...

Response: The user enjoys action-packed, high-stakes narratives with strong protagonists overcoming formidable odds. They prefer films that blend suspense, tension, and personal growth. There's a clear affinity for classic cinema...

Our goal is to find and remove items that, when excluded, significantly change this profile. The hypothesis is that filtering out inconsistent history items will help ACCENT produce CFEs more

Table 1: Statistics of the datasets.

Datasets	#users	#items	#interactions	Density
ML-1M	6,040	3,952	1,000,209	4.19%
Amazon	94,761	25,612	814,587	0.03%

aligned with true user preferences. First, we generate a textual profile t_u of the user’s preferences using an LLM prompted on their full interaction history and item descriptions:

$$t_u = \text{LLM}(\text{concat}(p_i))_{i \in H_u} \quad (2)$$

then we embed t_u into a high-dimensional vector e_u using Sentence Transformer (SBERT) [26]:

$$e_u = \text{SBERT}(t_u) \quad (3)$$

where p_i is description of item i . We start with the full interaction history $H_u^{(0)} = H_u$. Then, for a fixed number of steps n we iterate:

- For every remaining item $i' \in H_u^{(j-1)}$ (history before step j), compute the embedding obtained after its removal $e_{u \setminus \{i'\}}^{(j-1)}$.
- Select the item whose removal produces the largest cosine dissimilarity with the current profile:

$$i_j^* = \arg \max_{i' \in H_u^{(j-1)}} (1 - \cos(e_u^{(j-1)}, e_{u \setminus \{i'\}}^{(j-1)})) \quad (4)$$

- Permanently discard i_j^* : $H_u^{(j)} = H_u^{(j-1)} \setminus \{i_j^*\}$.

After n iterations, the set of all discarded items is $\tau_u = \{i_1^*, \dots, i_n^*\}$.

We can finally create the filtered history $H_u^{(n)} = H_u \setminus \tau_u$ and feed it to ACCENT (or any other history-based CFE framework), ensuring explanations exclude these misaligned items.

5 Experiments

This section outlines our experimental setup, including datasets, evaluation metrics, and baselines.

5.1 Datasets

We evaluate our approach using two datasets: MovieLens 1M (ML-1M) [12] and Amazon Video Games [14]. Both datasets provide user-item interactions and rich item metadata, including textual descriptions and categories. This metadata is used to create the LLM prompts as previously discussed. For the Amazon dataset, we ensure all users and items have at least five interactions. Table 1 shows statistical properties of these datasets.

To keep LLM costs manageable, we focus on users whose histories are most vulnerable to popularity bias. Following previous research [1], we rank all users on the fraction of popular items in their history. We retain only the extremes: the top 20% (*blockbuster* users) and bottom 20% (*niche* users), discarding the middle 60%. Users with more than 100 interactions are also dropped to curb runtime. From the survivors, we take the 250 most-niche and 250 most-blockbuster users per dataset to evaluate our approach¹.

¹Filtering steps were only applied for generating counterfactuals; the recommendation models were trained on the full datasets.

5.2 Baselines

To evaluate LLM-augmented ACCENT, we compare its counterfactual sets against two baselines:

- *ACCENT*. We use the original ACCENT method [32] as the primary baseline, as our framework builds directly upon it. We use $k = 5$, checking each of the top-5 recommendations for feasibility as a replacement for the top-1.
- *Top Popular*. This heuristic baseline constructs counterfactual sets from the most popular items in a user’s history, matching ACCENT’s set size per user. The expectation is that this performs poorly for non-blockbuster users.

5.3 Evaluation metrics

We report two complementary metrics to quantify how closely counterfactual explanations mirror the popularity profiles of the users who receive them.

Popularity Distribution Similarity (PDS). Let H be the complete collection of items that appear in users’ interaction histories and C the collection of items that appear in the generated counterfactual explanations. If an item occurs multiple times (e.g., across different users), it is included that many times in the collection. We convert each item to its global popularity score and place the scores into B equal-width bins, producing two histograms (empirical distributions) P_H and P_C . The similarity between the two popularity profiles can then be defined as the χ^2 distance:

$$\text{PDS} = \sum_{b=1}^B \frac{(P_H(b) - P_C(b))^2}{P_C(b) + \epsilon} \quad (5)$$

with $\epsilon = 10^{-10}$ for numerical stability. A smaller PDS indicates that counterfactual items follow nearly the same popularity distribution as the historical items, suggesting lower popularity bias.

Expected Popularity Deviation (EPD). EPD tracks the average shift in popularity at the user level. With $\bar{\pi}_{H_u}$ and $\bar{\pi}_{C_u}$ denoting the mean popularity of user u ’s history and counterfactual set respectively:

$$\text{EPD} = \frac{1}{|U|} \sum_{u \in U} (\bar{\pi}_{C_u} - \bar{\pi}_{H_u})^2 \quad (6)$$

Here we calculate the global EPD, but we can also use it locally for a single user by only computing $\text{EPD}_u = (\bar{\pi}_{C_u} - \bar{\pi}_{H_u})^2$. A small EPD indicates that a user receives counterfactual sets whose popularity closely matches what they historically prefer.

It should be noted that, as counterfactual sets are usually much smaller than the user history (often only one or two items), looking at the user level can yield volatile results. For this reason, we pool the users in PDS and examine the global EPD.

5.4 Implementation details

For our LLM-augmented CFEs, we employed Qwen3-8B, a recent open-weight language model. This 8 billion parameter instruction-tuned model, developed by Alibaba Group as part of the Qwen3 series, allows for both reasoning and traditional queries. In our experiments, we do not use the reasoning functionality. We use a quantized model (Q3_K_M) created by Unsloth [6] to improve performance. For inference, we used the llama-cpp-python library. The model was not fine-tuned on multimedia data.

Table 2: Popularity alignment comparison of methods with respect to item groups on ML-1M and Amazon datasets.

Dataset	Method	Niche		Blockbuster	
		PDS ↓	EPD ↓	PDS ↓	EPD ↓
ML-1M	Top Popular	1764.31	0.111	2588.44	0.070
	ACCENT	8.58	0.012	21.58	0.017
	LLM-Augmented	5.24	0.008	16.25	0.014
Amazon	Top Popular	158.88	$5.1 * 10^{-7}$	165.87	$4.2 * 10^{-5}$
	ACCENT	3.83	$2.0 * 10^{-8}$	14.29	$1.4 * 10^{-5}$
	LLM-Augmented	9.13	$1.9 * 10^{-8}$	6.06	$1.4 * 10^{-5}$

To compute the vector representations of the LLM-generated natural language profiles, we used Mixedbread’s mxbai-embed-large-v1 model [17]. This open-weight model achieves state-of-the-art performance in its size class. All embedding operations were performed using the SentenceTransformers (SBERT) [26]. For recommendation, we used a standard Neural Matrix Factorization (NeuMF) model [13]. The model was trained with an embedding size of 16, a learning rate of 1e-3, and a weight decay of 1e-3.

The implementation and hyperparameters of our recommendation model and counterfactual approach were based on the publicly available ACCENT codebase². We extended this baseline to LLM-augmented ACCENT. Our implementation can be found at <https://github.com/ahasami/llm-augmented-cfs>.

6 Results

We begin by asking a simple question: *does the LLM-augmentation step nudge counterfactual explanations towards the side of the popularity spectrum that each user actually favours?* To answer this, we generate ACCENT explanations with and without the filter. On ML-1M we allow the filter to drop $n = 5$ history items per user, whereas on Amazon we cap it at $n = 1$ so as not to wipe out the already thin behavioural signal. This dataset is a lot sparser than ML-1M, and a single deletion in a profile with few interactions can have a large impact. This sparsity also affects the performance of the recommendation model. We report nDCG@10 of 0.1303 and 0.0481 on ML-1M and Amazon respectively, computed using the negative sampling approach described in [13]. Here, for each test point of a user, 100 items the user has not interacted with are randomly sampled to form the candidate set.

Table 2 shows that our method achieves the best scores across all groups, consistently lowering both PDS and EPD compared to ACCENT. However, most differences are not statistically significant, except the overall EPD on ML-1M (blockbuster and niche combined, not shown). Our approach outperforms ACCENT (0.017 vs 0.014, $p < 0.05$).

Fig. 2a shows a clear, consistent shift on ML-1M. For bins of niche users (left panel) our filtered counterfactuals move downward and slightly rightward, indicating less-popular items drawn from deeper in each user’s long-tail history. For the bins of blockbuster users (right panel) the same mechanism pushes explanations upward and leftward, highlighting even more popular items near the head of the

profile. The result is a bidirectional correction. The same analysis on Amazon (Fig. 2b) is noisier. Some bins of blockbuster users benefit from a popularity boost, and for niche users the range of the user profile from which counterfactuals are generated is tightened.

We noticed a slight increase in cases where no CF could be found (ML-1M 14.56% → 21.16%, Amazon 27.09% → 27.31%). This makes sense, as we remove items that ACCENT can use to justify a change. By shrinking the candidate set, we sometimes eliminate the items that would have enabled a recommendation flip, especially for users whose profiles were minimal or highly concentrated.

Overall, the results show that LLM augmentation improves the popularity alignment of CFEs, especially on ML-1M, where user histories are richer and more stable. Bidirectional corrections are consistent: niche users get less popular explanations, while blockbuster users are nudged toward mainstream ones. Gains in EPD on ML-1M are small but statistically significant. On Amazon, impact is weaker—sparser histories limit the LLM’s ability to infer profiles and guide filtering. Still, the method never harms performance, suggesting it remains safe to apply even with limited effect.

7 Discussion

Limitations. While our LLM-augmentation step shows promising improvements in user alignment, several limitations remain. In sparse datasets, the deletion of even a single item can disproportionately distort user profiles, and the LLM may struggle to infer coherent preferences. The LLM itself also introduces challenges: it can be inconsistent, generic or even hallucinatory in its summaries. Although our prompt design mitigates some of these issues, occasional failures still occur. The approach is also computationally expensive, as we need $n * h$ LLM queries per user, where n is the number of iterations and h the size of the user history. Finally, removing out-of-character items can increase the number of instances where ACCENT cannot find a valid CFE.

Future work. Our method operates purely as a pre-processing step, without modifying the recommendation model or its outputs. Future work could look at how the recommendations would change, either by retraining or estimation through the usage of influence functions [16]. Also, exploring the interaction between LLM-augmented ACCENT and debiased recommendation outputs could offer insight into whether the observed biases in ACCENT stem from input data or the recommender itself. Finally, investigating the effects of larger models, or models fine-tuned on multimedia data, may further improve profile quality and filtering accuracy.

8 Conclusion

We introduced a straightforward, model-agnostic pre-processing step that uses large language models to improve the popularity alignment of counterfactual explanations in recommender systems. By filtering out history items inconsistent with a user’s core preferences, we guide the explanation process toward outputs more representative of users. We specifically examined the effect of our method on popularity bias and showed that the approach improves popularity alignment, particularly in richer datasets like MovieLens. While benefits are less profound for sparser datasets, the method does not degrade performance and provides a promising foundation for further research into bias-aware explanation using LLMs.

²<https://github.com/hieptk/accnt>

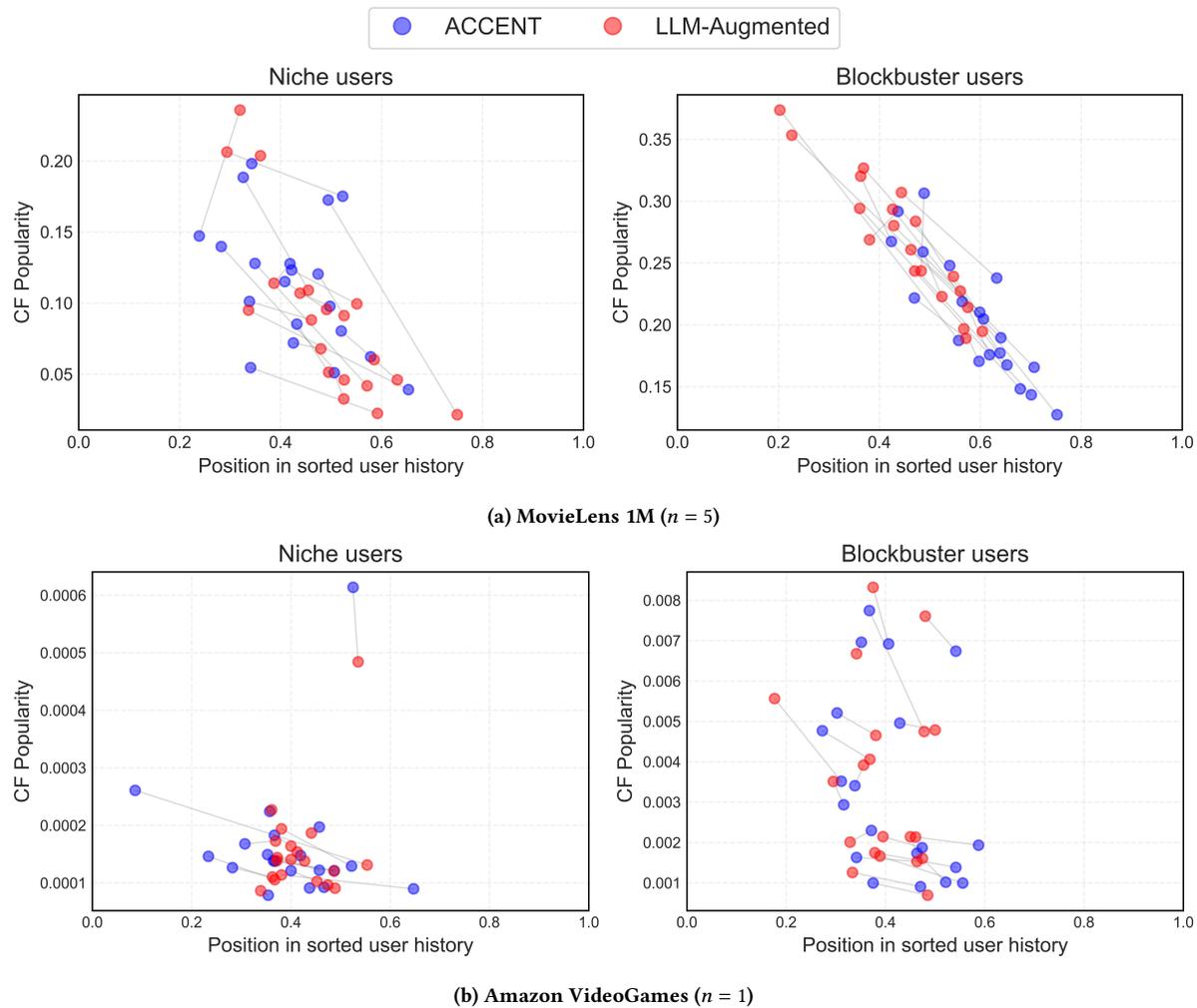


Figure 2: Average popularity of counterfactual sets versus average position of counterfactual items in the normalized popularity-sorted user history. Each user category is binned into 20 bins based on the popularity of similar histories.

References

- [1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The Unfairness of Popularity Bias in Recommendation. arXiv:1907.13286 [cs.LG] <https://arxiv.org/abs/1907.13286>
- [2] Martin Atzmueller, Johannes Fürnkranz, Tomáš Kliegr, and Ute Schmid. 2024. Explainable and interpretable machine learning and data mining. *Data Mining and Knowledge Discovery* 38, 5 (01 Sep 2024), 2571–2595. doi:10.1007/s10618-024-01041-y
- [3] Ziheng Chen, Fabrizio Silvestri, Jia Wang, Yongfeng Zhang, Zhenhua Huang, Hongshik Ahn, and Gabriele Tolomei. 2022. GREASE: Generate Factual and Counterfactual Explanations for GNN-based Recommendations. arXiv:2208.04222 [cs.LG] <https://arxiv.org/abs/2208.04222>
- [4] Weiyu Cheng, Yanyan Shen, Linpeng Huang, and Yanmin Zhu. 2019. Incorporating Interpretability into Latent Factor Models via Fast Influence Analysis. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 885–893. doi:10.1145/3292500.3330857
- [5] Henriette Cramer, Vanessa Evers, Satyan Ramml, Maarten van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. 2008. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction* 18, 5 (01 Nov 2008), 455–496. doi:10.1007/s11257-008-9051-3
- [6] Michael Han Daniel Han and Unslloth team. 2023. *Unslloth*. <http://github.com/unsllothai/unslloth>
- [7] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. 2020. Recommender Systems Leveraging Multimedia Content. *Comput. Surveys* 53, 5, Article 106 (Sept. 2020), 38 pages. doi:10.1145/3407190
- [8] Manqing Dong, Feng Yuan, Lina Yao, Xianzhi Wang, Xiwei Xu, and Liming Zhu. 2022. A survey for trust-aware recommender systems: A deep learning perspective. *Knowledge-Based Systems* 249 (2022), 108954. doi:10.1016/j.knsys.2022.108954
- [9] Yingqiang Ge, Shuchang Liu, Zuohui Fu, Juntao Tan, Zelong Li, Shuyuan Xu, Yunqi Li, Yikun Xian, and Yongfeng Zhang. 2024. A Survey on Trustworthy Recommender Systems. *ACM Trans. Recomm. Syst.* 3, 2, Article 13 (Nov. 2024), 68 pages. doi:10.1145/3652891
- [10] 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* (Houston, TX, USA) (WSDM '20). Association for Computing Machinery, New York, NY, USA, 196–204. doi:10.1145/3336191.3371824
- [11] Frank R. Hampel. 1974. The Influence Curve and its Role in Robust Estimation. *J. Amer. Statist. Assoc.* 69, 346 (1974), 383–393. doi:10.1080/01621459.1974.10482962 arXiv:<https://www.tandfonline.com/doi/pdf/10.1080/01621459.1974.10482962>
- [12] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages.

- doi:10.1145/2827872
- [13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (Perth, Australia) (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. doi:10.1145/3038912.3052569
- [14] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiushi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. arXiv:2403.03952 [cs.IR] <https://arxiv.org/abs/2403.03952>
- [15] Anastasiia Klimashevskaya, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. 2024. A survey on popularity bias in recommender systems. *User Modeling and User-Adapted Interaction* 34, 5 (2024), 1777–1834.
- [16] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (Sydney, NSW, Australia) (ICML'17)*. JMLR.org, 1885–1894.
- [17] Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. *Open Source Strikes Bread - New Fluffy Embedding Model*. <https://www.mixedbread.ai/blog/mxbai-embed-large-v1>
- [18] Jan Malte Lichtenberg, Alexander Buchholz, and Pola Schwöbel. 2024. Large language models as recommender systems: A study of popularity bias. <https://www.amazon.science/publications/large-language-models-as-recommender-systems-a-study-of-popularity-bias>
- [19] Ana Lucic, Maartje ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. 2022. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. arXiv:2102.03322 [cs.LG] <https://arxiv.org/abs/2102.03322>
- [20] Masoud Mansoury. 2022. Understanding and mitigating multi-sided exposure bias in recommender systems. *ACM SIGWEB Newsletter 2022*, Autumn (2022), 1–4.
- [21] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 2145–2148.
- [22] Giacomo Medda, Francesco Fabbri, Mirko Marras, Ludovico Boratto, and Gianni Fenu. 2024. GNNUERS: Fairness Explanation in GNNs for Recommendation via Counterfactual Reasoning. *ACM Trans. Intell. Syst. Technol.* 16, 1, Article 6 (Dec. 2024), 26 pages. doi:10.1145/3655631
- [23] Christoph Molnar. 2025. *Interpretable Machine Learning* (3 ed.). Independently published. <https://christophm.github.io/interpretable-ml-book>
- [24] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. 2019. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. *CoRR* abs/1905.07697 (2019). arXiv:1905.07697 <http://arxiv.org/abs/1905.07697>
- [25] Juno Prent and Masoud Mansoury. 2024. Correcting Popularity Bias in Recommender Systems via Item Loss Equalization. *arXiv preprint arXiv:2410.04830* (2024).
- [26] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).
- [27] Milad Sabouri, Masoud Mansoury, Kun Lin, and Bamshad Mobasher. 2025. Towards Explainable Temporal User Profiling with LLMs. In *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization (UMAP Adjunct '25)*. Association for Computing Machinery, New York, NY, USA, 219–227. doi:10.1145/3708319.3733655
- [28] J. Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce (Denver, Colorado, USA) (EC '99)*. Association for Computing Machinery, New York, NY, USA, 158–166. doi:10.1145/336992.337035
- [29] Ruoxi Shang, K. J. Kevin Feng, and Chirag Shah. 2022. Why Am I Not Seeing It? Understanding Users' Needs for Counterfactual Explanations in Everyday Recommendations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (Seoul, Republic of Korea) (FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 1330–1340. doi:10.1145/3531146.3533189
- [30] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. 2021. Counterfactual Explainable 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, 1784–1793. doi:10.1145/3459637.3482420
- [31] Nava Tintarev and Judith Masthoff. 2007. Effective explanations of recommendations: user-centered design. In *Proceedings of the 2007 ACM Conference on Recommender Systems (Minneapolis, MN, USA) (RecSys '07)*. Association for Computing Machinery, New York, NY, USA, 153–156. doi:10.1145/1297231.1297259
- [32] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual Explanations for Neural Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (Virtual Event, Canada) (SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1627–1631. doi:10.1145/3404835.3463005
- [33] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. arXiv:1711.00399 [cs.AI] <https://arxiv.org/abs/1711.00399>