Are We Explaining the Same Recommenders? Incorporating Recommender Performance for Evaluating Explainers

Amir Reza Mohammadi, Andreas Peintner, Michael Müller, Eva Zangerle
Department of Computer Science, University of Innsbruck
Innsbruck, Austria
amir.reza,andreas.peintner,michael.m.mueller,eva.zangerle@uibk.ac.at

ABSTRACT

Explainability in recommender systems is both crucial and challenging. Among the state-of-the-art explanation strategies, counterfactual explanation provides intuitive and easily understandable insights into model predictions by illustrating how a small change in the input can lead to a different outcome. Recently, this approach has garnered significant attention, with various studies employing different metrics to evaluate the performance of these explanation methods. In this paper, we investigate the metrics used for evaluating counterfactual explainers for recommender systems. Through extensive experiments, we demonstrate that the performance of recommenders has a direct effect on counterfactual explainers and ignoring it results in inconsistencies in the evaluation results of explainer methods. Our findings highlight an additional challenge in evaluating counterfactual explainer methods and underscore the need to report the recommender performance or consider it in evaluation metrics.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender Systems, Counterfactual Explanation, Evaluation

ACM Reference Format:
Amir Reza Mohammadi, Andreas Peintner, Michael Müller, Eva Zangerle . 2024. Are We Explaining the Same Recommenders? Incorporating Recommender Performance for Evaluating Explainers. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3640457.3691709

1 INTRODUCTION

Over the past decade, the rapid growth of deep learning models has driven remarkable progress in a wide range of fields, such as natural language processing [4], computer vision [12], and recommender systems (RecSys) [27, 28]. Addressing the need for transparency and interpretability, Explainable Artificial Intelligence (XAI) has garnered substantial interest across various communities. Among these approaches, counterfactual explainers (CE) [10] aim to advance model explainability. CE not only provides intuitive and easily understandable insights into model predictions [19] but also enables users to grasp how minor alterations in the input can lead to divergent outcomes. Counterfactual methods explore "what-if" scenarios to determine how changes in a user's data would affect the model's recommendations [13]. These methods are known for being user-friendly and easy to understand [13]. As a result, departing from conventional CE studies centered on tabular or image data, there is a growing emphasis on CE within RecSys [14, 18, 25, 29, 35, 38].

Various studies have addressed the challenge of evaluating counterfactual explainers across different domains, each considering their unique characteristics [7, 20, 22, 34]. Particularly, previous research has shown that model accuracy significantly impacts the performance of explainer models [7, 17, 22, 34] and should be reported for a consistent and reproducible research [8, 9, 36, 38]. We argue that this is a critical and often overlooked aspect in the RecSys community. We should evaluate the explainer model's ability to genuinely *explain* recommendations that are indeed of interest to the user (i.e., a high-performing recommender), rather than merely *justifying* uninteresting recommendations (i.e., a low-performing recommender). In other words, a good explainer should effectively explain the recommendations of a high-performing recommender model. This is especially important in the case of CE models, where the primary goal is to find the minimal change that results in a different recommendation. When the recommender is low-performing and therefore, lacks robustness, any change may lead to a new set of recommendations, making the explanation task less meaningful. The importance and effect of model performance on explainer performance have already been recognized in other communities [20] like Graph Neural Network-based models [7, 17, 22, 34] where explainer methods report the performance of the classifier models separately.

Recently, [8] presented metrics on CE for RecSys, although overlooked the importance of model performance. In our experiments, we demonstrate that the effectiveness of various explainer methods varies depending on the performance levels of the recommenders used. Since the performance of recommender models can vary significantly, we first show how this variability affects the performance of explainer models. Consequently, we argue for a metric that also considers model performance. In this paper, we utilize two types of recommenders, six explainer methods, and three real-world datasets to illustrate that evaluating explainer methods consistently, requires considering the performance of the recommenders.

2 BACKGROUND AND RELATED WORK

The field of interpretable ML has experienced notable advancements, particularly in the context of local interpretability, as seen in early works like LIME [32] and SHAP [23]. During this period, research on interpretability primarily focused on making the model itself more explainable [27]. Concurrently, the term "explainability" emerged,

RecSys '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0505-2/24/10.
https://doi.org/10.1145/3640457.3691709

referring to methods that treat the model as a black box and aim to retrospectively define its inner workings by so-called *post-hoc* explanations. Counterfactual explainers opened avenues for more nuanced explainer methods [7, 17, 22, 24, 34], enabling users to understand not only why a model made a specific prediction but also how different input configurations might alter the outcome.

For this study, we selected the most inclusive types of CE, specifically model-agnostic, post-hoc methods [37]. Model-agnostic methods are designed to be compatible with any differentiable recommender algorithm. Unlike explanation techniques specific to matrix factorization [2, 3], methods of our focus, can be used to explain any differentiable recommender. Although there are related works that use a similar approach in the context of CE for Graph Neural Network-based models [20], this is the first study to focus on evaluation metrics of CE for RecSys.

3 EVALUATION SETUP

We use the MovieLens 1M (ML-1M) [15], Yahoo! Music [11] and Pinterest [16] datasets, focusing on collaborative filtering models based on implicit user-item interactions [30]. To simulate implicit ratings in ML-1M, following [8], we retained only ratings of 3.5 or higher and included users and items with at least two ratings. This resulted in 575,128 ratings from 6,037 users across 3,381 items. For the Yahoo! Music dataset we retained ratings of 70 or higher, resulting in 19,155 users and 9,362 items. We split each dataset into training and testing (80/20) subsets using user identifiers. For each user, we generate a ranked list of recommendations and assess the explanation of the top recommendation.

We train our recommenders based on Hit Rate with cutoffs@10, 50, and 100. We categorize the perfromance of each recommender into three levels for comparing the effectiveness of explainer models. For the highest-performing category (*Gold*), we train the recommender for the full number of epochs. For the lowest-performing category (*Bronze*), we select the best recommender after 30% of the training epochs, and for the middle-performing category (*Silver*), we select the best recommender after 60% of the epochs. We also ensure that the performance hierarchy (from low to high), consistently shows at least a 20% performance gain between each category, otherwise we pick lower performing recommenders from each bin. For the VAE recommender on the ML-1M dataset, shown in Table 1, the *Gold* recommender achieved Hit Rates of 0.43 and 0.52, the *Silver* recommender achieved 0.34 and 0.48, and the *Bronze* recommender achieved 0.26 and 0.39 at cutoffs@10 and @50, respectively.

We hypothesize that if the ranking of explainer methods remains consistent across *Bronze*, *Silver*, *Gold* recommenders with varying performance levels, then recommender performance does not impact the evaluation of explainer methods. For instance, an explainer performing best with the *Bronze* recommender, but second best with the *Silver* recommender and worst among others with the *Gold* recommender, would confirm our hypothesis on the effect of recommender performance on explainer performance.

We evaluate 18 different configurations, repeat our experiments 3 times and report the average performance. We share our dataset processing scripts, the source code, and the hyper-parameters[1].

[1] https://github.com/dbis-uibk/CFX-Metric

3.1 Evaluated Methods

We examine a range of explanation methods, from solid baselines to recent approaches. Specifically, we have selected two similarity methods, two traditional explainers, and two state-of-the-arts methods to test our hypothesis:

Jaccard [6]/*Cosine* [33] Similarity generates explanations for an item by comparing it to items in the user's history using Jaccard [6] or Cosine [33] pairwise similarity calculated based on users who interacted with both items.

LIME-RS [26] is an adaptation of the LIME framework tailored for RecSys. It generates explanations by approximating the complex recommender model with a simple, interpretable linear model, focusing on a local neighborhood around the user's data.

SHAP (SHapley Additive exPlanations) [38] is a counterfactual method that calculates the contribution of each feature to the model's predictions. It uses game theory to assign a value to each feature based on its impact on the output.

ACCENT [36] is a CE framework that leverages influence functions to provide model-agnostic, actionable insights into why certain recommendations are made, extending the approach originally developed for latent factor models to a wider range of neural recommender systems.

LXR [8] represents the current state-of-the-art for CE for RecSys. It uses a self-supervised learning approach to create explanation masks that highlight the most influential user data for specific recommendations, without requiring perturbations.

We consider two collaborative filtering recommenders in our study. (1) *Matrix Factorization (MF)* decompose a user-item interaction matrix into lower-dimensional latent factors [1]. Recent reproducibility studies show that they (still) reach competitive results [31]. We have used the same implementation as [8] for easier comparison. (2) *Variational Autoencoder (VAE)* generative models that encode and decode data while learning a probabilistic latent representation [5]. They have been shown to be effective for collaborative filtering [21]. We included a VAE-based recommender with a similar architecture to [21].

3.2 Evaluated Metrics

We examine CE models using implicit user data, as a binary vector $x \in \{0,1\}^V$ indicating items the user has consumed [8]. Recommendations are given as ranked lists based on the recommender's predicted affinity scores, $f_\theta(x_u)[i]$, for each item i. We introduce perturbations by removing items from the user's vector x according to their *explainability* score from the explainer. In positive perturbation tests, items are removed in descending order of importance, expecting the explained item's score to drop and its rank to decrease. In negative perturbation tests, items are removed in ascending order of importance, expecting the explained item to maintain its high score and rank [8]. We employ stepwise perturbations where on each step $\frac{1}{M}$ of the user's data is deleted according to its explainer relevance score. M is a positive integer that serves as a granularity factor for the number of perturbation steps. Following [8], we set the value to $M = 10$. Let x_u represent user u's historical items vector. In a positive perturbation test, $x_u^{\text{pos}}(m)$ denotes user u's data after removing $\frac{m}{M}$ of the most important items according to the explainer. For negative perturbations, $x_u^{\text{neg}}(m)$ represents the data

Table 1: *AUC Values for Explaining Three Core Categories (VAE Recommender) on the ML-1M Dataset.* The best results in each evaluation are highlighted in bold, and the second-best results are *underlined*. Arrows next to the metrics indicate performance direction: a downward arrow (↓) signifies that lower values are better, while an upward arrow (↑) signifies that higher values are better. As shown in the table, the performance of the explainers varies across different recommender categories.

	Metric / Method	k = 5		k = 10		k = 20		DEL ↓	INS ↑	NDCG ↓
		POS ↓	NEG ↑	POS ↓	NEG ↑	POS ↓	NEG ↑			
Bronze Rec.	Jaccard [6]	0.718	**0.898**	0.816	0.945	0.885	0.966	0.0032	0.0095	0.652
	Cosine [33]	0.684	0.902	0.784	0.945	0.862	0.966	0.0032	0.0096	0.628
	LIME-RS [26]	0.711	0.916	0.809	**0.948**	0.880	**0.967**	0.0037	0.0071	0.655
	SHAP [38]	0.813	0.763	0.886	0.849	0.930	0.913	0.0043	0.0066	0.729
	ACCENT [36]	0.625	0.861	0.720	0.926	0.803	0.958	**0.0027**	**0.0110**	0.597
	LXR [8]	**0.587**	0.878	**0.686**	0.927	**0.773**	0.955	0.0029	0.0106	**0.564**
Silver Rec.	Jaccard [6]	0.380	0.715	0.461	**0.808**	0.554	0.872	0.0069	0.0186	0.413
	Cosine [33]	**0.368**	0.716	**0.449**	0.805	**0.541**	**0.868**	**0.0068**	0.0187	**0.407**
	LIME-RS [26]	0.444	**0.725**	0.540	0.798	0.635	0.848	0.0085	0.0153	0.459
	SHAP [38]	0.569	0.502	0.669	0.589	0.75	0.679	0.010	0.0115	0.548
	ACCENT [36]	0.426	0.538	0.507	0.667	0.595	0.769	0.0072	0.0168	0.460
	LXR [8]	0.427	0.670	0.515	0.747	0.607	0.804	0.0077	**0.0184**	0.451
Gold Rec.	Jaccard [6]	0.447	**0.842**	0.547	**0.892**	0.648	0.923	0.0066	0.0187	0.462
	Cosine [33]	**0.430**	0.841	**0.527**	0.888	**0.624**	**0.921**	**0.0065**	**0.0190**	**0.451**
	LIME-RS [26]	0.540	0.833	0.644	0.877	0.737	0.912	0.0077	0.0155	0.531
	SHAP [38]	0.670	0.607	0.757	0.701	0.830	0.783	0.0098	0.0118	0.625
	ACCENT [36]	0.515	0.668	0.597	0.784	0.686	0.862	0.0066	0.0177	0.530
	LXR [8]	0.604	0.589	0.696	0.675	0.776	0.761	0.0077	0.0140	0.058

after removing $\frac{m}{M}$ of the least important items. In both tests, we perform $m = 1, \ldots, M$ steps, gradually deleting items in decreasing or increasing order of importance. The rank of the explained item for user u is denoted by $\text{rank}(x_u)$. Given these notations, we evaluate the explainer models' performance by measuring the area under the curve (AUC) in following counterfactual perturbation tests:

(1) Positive Perturbations@K (POS-P@K) measures how quickly the explained item falls out of the top K recommendations during a positive perturbation test. For each step m, with $\mathbb{1}[\cdot]$ as the indicator function, it is defined as

$$\text{POS}-P@K(m) = \mathbb{1}\left[\text{rank}\left(\mathbf{x}_u^{pos}(m)\right) \leq K\right]. \tag{1}$$

(2) Negative Perturbations@K (NEG-P@K) assesses if the explained item remains in the top K recommendations in a negative perturbation test. It is represented as

$$\text{NEG}-P@K(m) = \mathbb{1}\left[\text{rank}\left(\mathbf{x}_u^{neg}(m)\right) \leq K\right]. \tag{2}$$

(3) Deletion Perturbations (DEL-P) evaluates how the recommender's score for the explained item decreases as the most crucial user data is removed. It is computed as

$$\text{DEL-P@K(m)} = f\left(\mathbf{x}_u^{pos}(m)\right)[i]. \tag{3}$$

(4) Insertion Perturbations (INS-P) assesses how the recommender's confidence improves as the most important user data is gradually added to an initially empty user vector. It is similar to a negative perturbation test, but in reverse. The INS-P is defined as

$$\text{INS-P@K(m)} = f\left(\mathbf{x}_u^{neg}(M-m)\right)[i]. \tag{4}$$

(5) NDCG Perturbations (NDCG-P) uses the Normalized Discounted Cumulative Gain (NDCG) in a positive perturbation test. It evaluates how quickly the explained item drops in ranking as the most important user data is gradually removed. For each step m,

$$\text{NDCG-P(m)} = \frac{1}{\log_2\left(1 + \text{rank}\left(\mathbf{x}_u^{pos}(m)\right)\right)}. \tag{5}$$

For POS-P, DEL-P, and NDCG-P, lower AUC values indicate better performance, showing the explained item drops quickly when key data is removed. Conversely, for NEG-P and INS-P, higher AUC values are preferable, indicating increased confidence in the explained item as important data is added. The metrics above are defined for a single user. We report our evaluations below based on the average value over a hidden test set of users.

4 EXPERIMENTS AND RESULTS

Our study aims to determine whether recommender performance should be considered when evaluating explainer methods for RecSys. To test our hypothesis regarding the consistency of explainer methods ranking, we train VAE and MF recommenders on three levels of performance, explain them using six explainer methods, and evaluate their performance with five explainer metrics. The full experimental results are available in our Git repository.

In Table 1, we present the performance of the explainer methods for the VAE recommender on the ML-1M dataset. As shown, LXR and ACCENT are the best-performing methods in the POS@k, DEL, INS, and NDCG metrics for the *Bronze* recommender. However,

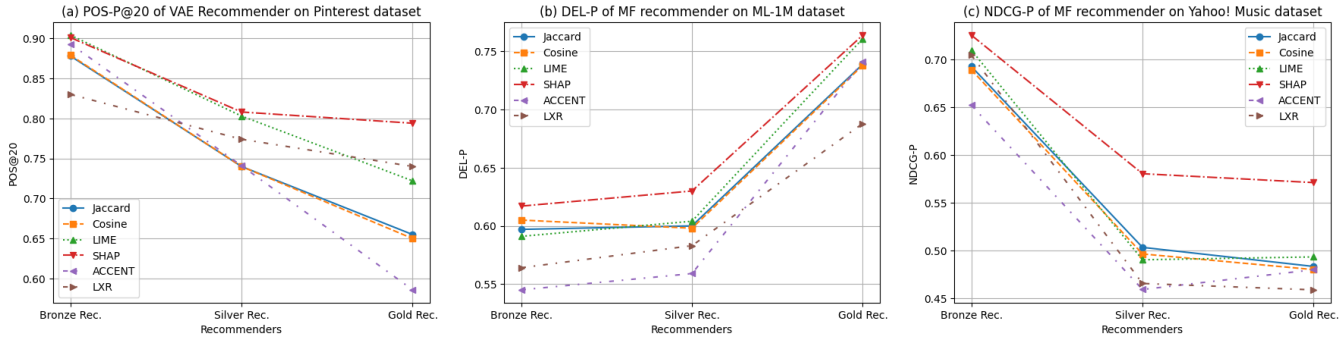

Figure 1: Performance summary of explainers across various metrics and datasets. Each figure illustrates the change in the best-performing explainers across three levels of recommender performance. For all metrics, lower values indicate better performance.

this diminishes for the *Silver* recommender and for the *Gold* recommender, similarity-based methods outperform others in most metrics. This shows that the performance of explainer methods is inconsistent across these levels. Specifically, the ranking of the best-performing explainers varies significantly. For instance, LXR outperforms ACCENT by 4.2% in POS@10 with the Bronze recommender. However, Cosine surpasses LXR by 8% with the Silver recommender and by 17% with the Gold recommender.

We repeat the same experiment with other recommenders and datasets and observe a similar picture. A summary of these results is illustrated in Figure 1, showing how different explainers perform. For instance, in Figure 1(a) we compare methods on POS@20 using VAE recommender in Pinterest dataset, where we observe that Jaccard and LXR outperform others on *Bronze* recommender, then Jaccard and Cosine are leading methods in *Silver* recommender and ACCENT is close to best performance. Then ACCENT keeps getting better and outperforms other methods on the *Gold* Recommender. In Figure 1(b) we compare methods based on DEL using MF recommender on ML-1M dataset, where we observe that ACCENT outperforms on *Bronze* recommender, but LXR outperforms in *Gold* recommender. We can also see the difference in the performance of the non-leading methods where Jaccard and Cosine are on par with ACCENT on *Gold* recommender, in contrast to their performance on the *Bronze* recommender. Also in Figure 1(c) we compare methods based on NDCG using MF recommener for Yahoo! Music dataset, where we observe that ACCENT outperforms on *Bronze* recommender but LXR outperforms in *Gold* recommender. When comparing the performance levels of explainers within the VAE recommender system, it is informative to look beyond just the top-performing models. Specifically, we observe that the rank of explainers changes 43.1% of the time when moving from a Bronze to a Silver recommender, 31.1% of the time when moving from Silver to Gold recommender, and 67.4% of the time when moving directly from Bronze to Gold recommender. This experiment demonstrates the intuitive effect of recommender quality on the performance of explainer methods.

5 CONCLUSION AND FUTURE WORK

We investigated the impact of recommender performance on the efficacy of counterfactual explainer (CE) methods within the domain of RecSys. Our analysis included two types of recommenders, six explainer methods, and three real-world datasets to illustrate our findings.

Our experiments revealed that the performance of the recommender significantly influences the effectiveness of explainer methods. Specifically, high-performing recommenders yielded different results in terms of ranking and quality of explainers compared to low-performing ones. It is essential to emphasize that our objective is neither to compare these methods nor to imply that one outperforms the others. The purpose of our experiments is not to benchmark or evaluate the relative performance of these methods. Our findings underscores the necessity of considering the performance of the recommender system when evaluating and comparing CE methods. Ignoring this factor can lead to misleading conclusions about the capabilities of explainer methods. Therefore, we propose that future research and evaluations in this field should adopt a metric that includes the performance of the underlying recommender, to ensure more consistent and reproducible results or at least report the recommender performance separately. We note that our primary objective is not to compare and benchmark various explainer methods, but rather to illustrate how the performance of recommender systems influences these methods. As a direction for future work, we suggest comprehensive benchmarking of current CE methods under standardized experimental settings to provide a clearer comparison of their capabilities. By incorporating these advancements, we can achieve more accurate assessments and drive further progress in explainable AI within RecSys, ultimately enhancing user trust and understanding of AI-driven recommendations.

REFERENCES

[1] Mohamed Hussein Abdi, George Onyango Okeyo, and Ronald Waweru Mwangi. 2018. Matrix Factorization Techniques for Context-Aware Collaborative Filtering Recommender Systems: A Survey. *Comput. Inf. Sci.* 11, 2 (2018), 1–10. https://doi.org/10.5539/CIS.V11N2P1

[2] Behnoush Abdollahi and Olfa Nasraoui. 2016. Explainable Matrix Factorization for Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*. ACM, 5–6. https://doi.org/10.1145/2872518.2889405

[3] Behnoush Abdollahi and Olfa Nasraoui. 2017. Using Explainability for Constrained Matrix Factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*. ACM, 79–83. https://doi.org/10.1145/3109859.3109913

[4] Keivan Alizadeh, Iman Mirzadeh, Dmitry Belenko, Karen Khatamifard, Minsik Cho, Carlo C. Del Mundo, Mohammad Rastegari, and Mehrdad Farajtabar. 2023. LLM in a flash: Efficient Large Language Model Inference with Limited Memory.

CoRR abs/2312.11514 (2023). https://doi.org/10.48550/ARXIV.2312.11514 arXiv:2312.11514

[5] Bahare Askari, Jaroslaw Szlichta, and Amirali Salehi-Abari. 2021. Variational Autoencoders for Top-K Recommendation with Implicit Feedback. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 2061–2065. https://doi.org/10.1145/3404835.3462986

[6] Sujoy Bag, Sri Krishna Kumar, and Manoj Kumar Tiwari. 2019. An efficient recommendation generation using relevant Jaccard similarity. *Inf. Sci.* 483 (2019), 53–64. https://doi.org/10.1016/J.INS.2019.01.023

[7] Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. 2021. Robust Counterfactual Explanations on Graph Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 5644–5655. https://proceedings.neurips.cc/paper/2021/hash/2c8c3a57383c63caef6724343eb62257-Abstract.html

[8] Oren Barkan, Veronika Bogina, Liya Gurevitch, Yuval Asher, and Noam Koenigstein. 2024. A Counterfactual Framework for Learning and Evaluating Explanations for Recommender Systems. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*. ACM, 3723–3733. https://doi.org/10.1145/3589334.3645560

[9] 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. *CoRR* abs/2208.04222 (2022). https://doi.org/10.48550/ARXIV.2208.04222 arXiv:2208.04222

[10] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. arXiv:1702.08608 [stat.ML]

[11] Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. 2012. The Yahoo! Music Dataset and KDD-Cup '11. In *Proceedings of KDD Cup 2011 competition, San Diego, CA, USA, 2011 (JMLR Proceedings, Vol. 18)*. JMLR.org, 8–18. http://proceedings.mlr.press/v18/dror12a.html

[12] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, Ranjay Krishna, Dustin Schwenk, Eli VanderBilt, and Aniruddha Kembhavi. 2023. Imitating Shortest Paths in Simulation Enables Effective Navigation and Manipulation in the Real World. *CoRR* abs/2312.02976 (2023). https://doi.org/10.48550/ARXIV.2312.02976 arXiv:2312.02976

[13] Timo Freiesleben. 2022. The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples. *Minds Mach.* 32, 1 (2022), 77–109. https://doi.org/10.1007/s11023-021-09580-9

[14] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*. ACM, 196–204. https://doi.org/10.1145/3336191.3371824

[15] 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. https://doi.org/10.1145/2827872

[16] 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, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 173–182. https://doi.org/10.1145/3038912.3052569

[17] Zexi Huang, Mert Kosan, Sourav Medya, Sayan Ranu, and Ambuj K. Singh. 2023. Global Counterfactual Explainer for Graph Neural Networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*. ACM, 141–149. https://doi.org/10.1145/3539597.3570376

[18] Neham Jain, Vibhhu Sharma, and Gaurav Sinha. 2024. Counterfactual Explanations for Visual Recommender Systems. In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*. ACM, 674–677. https://doi.org/10.1145/3589335.3651484

[19] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic Recourse: from Counterfactual Explanations to Interventions. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*. ACM, 353–362. https://doi.org/10.1145/3442188.3445899

[20] Mert Kosan, Samidha Verma, Burouj Armgaan, Khushbu Pahwa, Ambuj K. Singh, Sourav Medya, and Sayan Ranu. 2023. GNNX-BENCH: Unravelling the Utility of Perturbation-based GNN Explainers through In-depth Benchmarking. *CoRR* abs/2310.01794 (2023). https://doi.org/10.48550/ARXIV.2310.01794 arXiv:2310.01794

[21] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. ACM, 689–698. https://doi.org/10.1145/3178876.3186150

[22] Ana Lucic, Maartje A. ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. 2022. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*. PMLR, 4499–4511. https://proceedings.mlr.press/v151/lucic22a.html

[23] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4765–4774. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[24] Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. 2022. CLEAR: Generative Counterfactual Explanations on Graphs. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. http://papers.nips.cc/paper_files/paper/2022/hash/a69d7f3a1340d55c720e572742439eaf-Abstract-Conference.html

[25] Amir Reza Mohammadi. 2023. Explainable Graph Neural Network Recommenders; Challenges and Opportunities. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 1318–1324. https://doi.org/10.1145/3604915.3608875

[26] Caio Nóbrega and Leandro Balby Marinho. 2019. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*. ACM, 1671–1678. https://doi.org/10.1145/3297280.3297443

[27] Andreas Peintner, Amir Reza Mohammadi, and Eva Zangerle. 2023. SPARE: Shortest Path Global Item Relations for Efficient Session-based Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 58–69. https://doi.org/10.1145/3604915.3608768

[28] Hossein A. Rahmani, Mohammadmehdi Naghiaei, and Yashar Deldjoo. 2024. A Personalized Framework for Consumer and Producer Group Fairness Optimization in Recommender Systems. *Trans. Recomm. Syst.* 2, 3 (2024), 19:1–19:24. https://doi.org/10.1145/3651167

[29] Niloofar Ranjbar, Saeedeh Momtazi, and MohammadMehdi Homayoonpour. 2024. Explaining recommendation system using counterfactual textual explanations. *Mach. Learn.* 113, 4 (2024), 1989–2012. https://doi.org/10.1007/S10994-023-06390-1

[30] Steffen Rendle. 2021. Item Recommendation from Implicit Feedback. *CoRR* abs/2101.08769 (2021). arXiv:2101.08769 https://arxiv.org/abs/2101.08769

[31] Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*. ACM, 240–248. https://doi.org/10.1145/3383313.3412488

[32] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 1135–1144. https://doi.org/10.1145/2939672.2939778

[33] Ramni Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula, and Gaurav Srivastav. 2020. Movie Recommendation System using Cosine Similarity and KNN. *International Journal of Engineering and Advanced Technology* 9 (06 2020), 2249–8958. https://doi.org/10.35940/ijeat.E9666.069520

[34] Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. 2022. Learning and Evaluating Graph Neural Network Explanations based on Counterfactual and Factual Reasoning. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. ACM, 1018–1027. https://doi.org/10.1145/3485447.3511948

[35] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. 2021. Counterfactual Explainable Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 1784–1793. https://doi.org/10.1145/3459637.3482420

[36] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual Explanations for Neural Recommenders. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 1627–1631. https://doi.org/10.1145/3404835.3463005

[37] Sahil Verma, John P. Dickerson, and Keegan Hines. 2020. Counterfactual Explanations for Machine Learning: A Review. *CoRR* abs/2010.10596 (2020). arXiv:2010.10596 https://arxiv.org/abs/2010.10596

[38] Jinfeng Zhong and Elsa Negre. 2022. Shap-enhanced counterfactual explanations for recommendations. In *SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, April 25 - 29, 2022*. ACM, 1365–1372. https://doi.org/10.1145/3477314.3507029