Explainable Fairness in Recommendation

Yingqiang Ge[1], Juntao Tan[1], Yan Zhu[2], Yinglong Xia[2], Jiebo Luo[3]
Shuchang Liu[1], Zuohui Fu[1], Shijie Geng[1], Zelong Li[1], Yongfeng Zhang[1]
[1]Rutgers University [2]Meta Platforms, Inc. [3]University of Rochester
{yingqiang.ge,juntao.tan}@rutgers.edu,{yzhu,yxia}@fb.com,jluo@cs.rochester.edu
{shuchang.liu,zuohui.fu,shijie.geng,zelong.li,yongfeng.zhang}@rutgers.edu

ABSTRACT

Existing research on fairness-aware recommendation has mainly focused on the quantification of fairness and the development of fair recommendation models, neither of which studies a more substantial problem–the identification of the source of model disparities in recommendation. This information is critical for the recommender system designers to understand the intrinsic recommendation mechanism and provides insights on how to improve model fairness to the decision makers. Fortunately, with the rapid development of explainable AI, we are able to use model explainability to gain insights into model (un)fairness. In this paper, We study the problem of explainable fairness in recommendation as we believe this type of study would motivate and guide the reasoning of fair recommender systems with a more promising and unified methodology. Particularly, we focus on a common setting with feature-aware recommendation and popularity bias. We propose a counterfactual explainable fairness framework, called CEF, which generates explanations that are able to improve fairness without significantly hurting the performance. The CEF framework formulates an optimization problem to learn the "minimal" change to a given feature that changes the recommendation results to a certain level of fairness. Based on the counterfactual recommendation result of each feature, we calculate an explainability score in term of the fairness-utility trade-off to rank all the feature-based explanations, and select the top ones as fairness explanations. Experimental results on several real-world datasets validate that our method is able to effectively provide explanations to the model disparities and these explanations can achieve better fairness-utility trade-off when using them for recommendation than all the baselines.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Explainable Fairness; Recommender Systems; Explainable Recommendation; Counterfactual Reasoning

ACM Reference Format:
Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, Yongfeng Zhang. 2022. Explainable Fairness in Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Nowadays, with the extensive deployment in e-commerce platforms, recommender systems (RS) have been widely acknowledged for their strong capabilities of delivering high-quality services to consumers [9, 20, 37, 41]. Despite these huge benefits, the issue of fairness in recommendation has also attracted considerable interests from both academia and industry [21, 29, 40]. Fortunately, these concerns about algorithmic fairness have resulted in a resurgence of interest to develop fairness-aware recommendation models to ensure that such models would not become a source of unfair discrimination in recommendation [14, 19, 28, 29]. In the area of fairness-aware recommendation, existing research mainly focuses on the quantification of fairness and the development of fair recommendation models. Fairness quantification aims to develop and investigate quantitative metrics that measure algorithmic disparities in ranking or recommendation [14, 16, 27]. For example, [14, 27] proposed and studied the recommendation quality unfairness between active users and inactive users. Meanwhile, fair recommendation aims to find feasible algorithmic approaches that can adjust the recommendation results to reduce recommendation disparities. For example, [2, 19] proposed approaches to mitigating the popularity bias between different item groups.

Despite the great efforts on fairness-aware recommendation and possibly countless future emergence of discoveries, one fundamental question that has not been studied extensively yet is fairness diagnostics, i.e.,

- **RQ** *What are the sources that result in model disparities in recommendation?*

Considering the huge commercial and social values that RS brought the e-commerce platforms and the society, we believe that the answer to this **RQ** is critical for recommendation system designers to understand the intrinsic recommendation mechanism and to provide insights for decision makers on how to improve model fairness. Yet, the answer to this question turns out to be unsurprisingly challenging especially when the predictive model is a large-scale deep black-box model with large number of input features. For example, it is hard to tell how input features (like screen size, battery, camera) would influence the popularity bias. Note that some pioneer works in other areas have leveraged explainable AI to seek for feature-based explanations for certain fairness outcome. For

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

instance, Begley et al. used Shapley value to attribute the model disparity in classification [4, 36]. Though their methods successfully provide explanations to the model disparities in simple tasks, they are not suitable for recommender systems where the model inputs could be extremely large and sparse. Furthermore, existing methods only partially answers the above question, since they only explain either utility or fairness alone, ignoring the fact that there is an inherent trade-off between fairness and utility, which has been demonstrated by several recent work both empirically and theoretically [24, 25, 30, 51]. And this incomplete view may potentially downgrade the stringency of the method because explanations that have the same effect on model fairness may not have the same effect on model utility.

In this paper, we propose a novel framework to explain the recommendation (un)fairness based on a counterfactual reasoning paradigm. Particularly, we focus on a common setting with feature-aware recommendation and item popularity bias [1, 2, 19]. Specifically, we propose a counterfactual explainable fairness framework (CEF) to generate feature-based explanations in terms of item exposure disparity for various black-box feature-aware recommendation models. We first follow prior works [11, 45, 54] to build a user-feature attention matrix as well as an item-feature quality matrix, and use both matrices to train a feature-aware recommendation model. Then, we aim to find the "minimum" changes to a given feature in the feature space that switch the recommendation results to a certain level of fairness. In order to avoid overwhelmed sacrificing of the recommendation quality, we also constrain the feature perturbation within a certain degree in the objective function. With the counterfactual learning objective and the perturbation constraint, our proposed framework is able to generate feature-level explanations that consider the fairness-utility trade-off. Finally, we calculate an explainability score in term of the fairness-utility trade-off based on the counterfactual recommendation result of each feature. These scores help ranking the feature-based explanations and we select the top ones as fairness explanations for the pre-trained recommendation model.

In general, the contributions of this work can be summarized as follows:

- We study the problem of explainable fairness in recommendation and propose a framework based on counterfactual reasoning. To the best of our knowledge, this is the first work that introduces explainable fairness into modern recommender systems.
- We design a learning-based intervention method to discover critical features that will significantly influence the fairness-utility trade-off and use them as fairness explanations for black-box feature-aware recommendation systems;
- We conduct extensive experiments to evaluate our frameworks's effectiveness and validate that explanations generated by CEF can achieve better fairness-utility trade-off when using them for recommendation than all the baselines.

2 RELATED WORK

There are several main research lines related to our work: explainable recommendation, fairness in recommendation and fairness explanation. We will briefly introduce each of them in this section.

2.1 Explainable Recommendation

Explainable recommendation has been an important topic in both academics and industry [53]. Recently, there are several promising methods proposed, as they help to improve the transparency, user satisfaction and trust over the recommender systems [53]. Early approaches mainly attempt to make latent factor models explainable by aligning each latent dimension with an explicit meaning, such as item features [12, 23, 55] or word topics [31]. Recently, with the ever prospering of deep learning technology, many neural algorithms are developed to explain recommendations based on neural models such as attention mechanism. For example, [38] proposed to attentively highlight particular words in user reviews as explanations, [8] proposed to retrieve informative sentences from reviews as explanations, and [13] proposed visually explainable recommendation to highlight image regions as explanations. In addition to text-based or image-based explainable recommendation, knowledge-aware explainable recommendation has also attracted research attention recently, such as [3, 44, 46].

Works using counterfactual reasoning to improve recommendation explainability [22, 41, 42, 47] have been proposed very recently. Ghazimatin et al. [22] tried to generate provider-side counterfactual explanations by looking for a minimal set of user's historical actions (e.g. reviewing, purchasing, rating) such that the recommendation can be changed by removing the selected actions. Xu et al.[47] proposed to improve this by using perturbation model to obtain counterfactuals. Tran et al. [42] adopted influence functions for identifying training points most relevant to a recommendation while deducing a counterfactual set for explanations. Tan et al.[41] proposed to generate explanations that considers the causal relations to the outcome.

Yet, our work is different from prior works on two key points: 1) In terms of problem definition, prior works generate counterfactual explanations to explain user behaviors or recommendation results, while our method generates such explanations to explain the fairness-utility trade-off in recommendation. 2) In terms of technique, our method adopts a counterfactual reasoning framework from a global perspective, which explains the entire model behavior, while prior works focus on generating individual explanations for an individual recommendation result.

2.2 Fairness in Recommendation

The issue of fairness in recommendation has received growing concerns as the recommender systems touch and influence more and more people in their daily lives. Several recent works focusing on fairness quantification have found various types of bias in recommendations, such as gender and race [2, 10], item popularity [1, 19], user feedback [14, 27, 28] and opinion polarity [49]. Meanwhile, the relevant methods for fair recommendation can be roughly divided into three categories: pre-processing, in-processing and post-processing algorithms [18, 29]. First of all, pre-processing methods usually aim to minimize the bias in the data sources. It includes fairness-aware sampling methodologies in the data collection process to cover items of all groups, balancing methodologies to increase coverage of minority groups, and repairing methodologies to ensure label correctness [18]. Secondly, in-processing methods aim at encoding fairness as part of the objective function, typically as a regularizer [1, 5]. Finally, post-processing methods tend to modify the presentations of the results, e.g., re-ranking through linear programming [27, 40, 48] or multi-armed bandit [7]. Based on the characteristics of the recommender system itself, there also have been a few works related to multi-sided fairness in multi-stakeholder systems [6, 17, 33].

Moreover, there are two primary paradigms adopted in recent studies on algorithmic discrimination: individual fairness and group fairness [17, 29]: individual fairness requires that each similar individual should be treated similarly; and group fairness requires that the protected groups should be treated similarly to the advantaged group or the populations as a whole. In this paper, we mainly focus on the item popularity fairness, which is a kind of group fairness and aims to achieve fair chance of exposure for different item groups [1, 2, 19].

2.3 Fairness Explanation

Explainability and fairness are two important perspectives for responsible recommender systems, however, the joint effort of these two is still less explored. There have been several pioneering studies trying to derive explanations for model fairness [4, 36] in other tasks. For example, Begley et al. [4] leveraged Shapley value paradigm [39] to attribute the feature contributions to model disparity to generate explanations. It estimates the sum of individual contributions from input features, so as to understand which feature contributes more to the model disparity [4]. Though this type of methods successfully provide explanations to the model disparities, they are not suitable for recommender systems. First of all, the definition of Shapley value is the average marginal contribution of a feature value across all possible coalitions, meaning that the computation time increases super-exponentially with the number of features. In recommendation systems, this becomes impractical since it is very common to have a large number of user/item features in the feature space. Secondly, the Shapley value can only explain either utility or fairness alone [4, 36], but not the fairness-utility trade-off. However, our proposed counterfactual explainable fairness framework (CEF) is able to mitigate the above problems.

3 EXPLAINABLE FAIRNESS

In this section, we first introduce how to use review information to generate user-feature matrix and item-feature matrix, then introduce the details of feature-aware recommendation systems. We introduce how to generate counterfactual explanations for fairness in section 3.4 and 3.5.

3.1 Feature Generation

Suppose we have a user set with m users denoted as $\mathcal{U}$, an item set $\mathcal{V}$ with n items and their interaction set $\mathcal{T} = \{(u, v) | u \in \mathcal{U}, v \in \mathcal{V}, u \text{ has interacted with } v\}$. Based on an open source toolkit for phrase-level sentiment analysis, called "Sentires"[1], we can easily convert the raw review information into a set of quadruples $\mathcal{W} = \{(u_l, v_l, f_l, s_l)\}_{l=1}^{N}$. Specially, each element $(u_l, v_l, f_l, s_l) \in \mathcal{W}$ means user $u_l \in \mathcal{U}$ mentioned feature $f_l \in \mathcal{F}$ of item $v_l \in \mathcal{V}$ with sentiment $s_l \in \mathcal{S}$, where $\mathcal{F}$ denotes the set of all features with size r and the sentiment set $\mathcal{S} = \{positive(+1), negative(-1)\}$. For example, in the review of "I like the color of this sweater, but the sleeve is not satisfied, since it is too tight for me.", the features are "collar" and "sleeve", and the user expresses positive and negative sentiments on them. The final extracted tuples are "(user, item, color, positive)" and "(user, item, sleeve, negative)", respectively. Following the same method described in [11, 41, 55], we construct a user-feature attention matrix $\boldsymbol{A} \in \mathbb{R}^{m \times r}$ and a item-feature quality matrix $\boldsymbol{B} \in \mathbb{R}^{n \times r}$ using all the quadruples in $\mathcal{W}$, where $\boldsymbol{A}_{u,f}$ indicates to what extent the user u cares about the feature f, and $\boldsymbol{B}_{v,f}$ indicates how well the item v performs on the feature f. Specifically, $\boldsymbol{A}$ and $\boldsymbol{B}$ are calculated as:

$$\begin{aligned}
\boldsymbol{A}_{u,f} &= \begin{cases} 0, \text{ if user } u \text{ did not mention feature } f \\ 1 + (M-1)\left(\frac{2}{1+\exp(-t_{u,f})} - 1\right), \text{ else} \end{cases} \\
\boldsymbol{B}_{v,f} &= \begin{cases} 0, \text{ if item } v \text{ has no review on feature } f \\ 1 + \frac{M-1}{1+\exp(-t_{v,f} \cdot \bar{t}_{v,f})}, \text{ else} \end{cases}
\end{aligned} \tag{1}$$

where M is the rating scale in the system, which equals to 5 (stars) in most cases, $t_{u,f}$ is the frequency that user u mentioned aspect f, $t_{v,f}$ is the frequency that item v is mentioned on feature f, and $\bar{t}_{v,f}$ is the average sentiment of these mentions. For both $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices, their elements are re-scaled into the range of $(1, M)$ using the sigmoid function (see Eq.(1)) to match with the original system's rating scale. Readers may refer to [55, 56] for more details and the same user-feature and item-feature matrix construction technique can also be found in [15, 26, 41, 45].

3.2 Feature-aware Recommender Systems

Once given the user-feature attention matrix $\boldsymbol{A}$ and item-feature quality matrix $\boldsymbol{B}$, we define a ranking model g that predicts the user-item ranking score $\hat{y}_{i,j}$ for user u_i and item v_j by:

$$\hat{y}_{u,v} = g(\boldsymbol{A}_u, \boldsymbol{B}_v \mid Z, \Theta) \tag{2}$$

where $\boldsymbol{A}_u$ and $\boldsymbol{B}_v$ are the vector of user u and the vector of item v, Θ is the model parameter, and Z represents all other auxiliary information. Depending on the application, Z could be rating scores, clicks, text, images, etc., and is optional in the recommendation model g.

In this work, we explore different implementations of g to demonstrate the effectiveness of our proposed framework. The general architecture of g is a multi-layer neural network, that is:

$$\hat{y}_{uv} = \boldsymbol{W}_T \sigma_T \left(\ldots \left(\boldsymbol{W}_1 \sigma_1 \left(merge\left(\boldsymbol{A}_u, \boldsymbol{B}_v\right)\right) + \boldsymbol{b}_1 \right) + \ldots \right) + \boldsymbol{b}_T \tag{3}$$

where, for the t th layer $(t \in [1, T]), \sigma_t$ is a non-linear activation function, $\boldsymbol{W}_t$ and $\boldsymbol{b}_t$ represent weights and bias, respectively. $merge(\cdot)$ is an operator merging the user-feature and item-feature vectors, and we explore it within the following functions:

- **Element-wise Product Merge**:

$$merge\left(\boldsymbol{A}_u, \boldsymbol{B}_v\right) = \boldsymbol{W}_U \boldsymbol{A}_u^T \odot \boldsymbol{W}_V \boldsymbol{B}_v^T. \tag{4}$$

  where $\boldsymbol{W}_U$ and $\boldsymbol{W}_V$ are trainable parameters, and $\odot$ represents the element-wise product (a.k.a. Hadamard product).

- **Concatenation Merge**:

$$merge\left(\boldsymbol{A}_u, \boldsymbol{B}_v\right) = [\boldsymbol{W}_U \boldsymbol{A}_u^T, \boldsymbol{W}_V \boldsymbol{B}_v^T]. \tag{5}$$

  where $\boldsymbol{W}_U$ and $\boldsymbol{W}_V$ are trainable parameters.

[1] https://github.com/evison/Sentires

Then, we train the model with a cross-entropy loss:

$$
\begin{aligned}
Loss &= -\sum_{u,v,y_{u,v}=1} \log \hat{y}_{u,v} - \sum_{u,v,y_{u,v}=0} \log(1-\hat{y}_{u,v}) \\
&= -\sum_{u,v} y_{u,v} \log \hat{y}_{u,v} + (1-y_{u,v}) \log(1-\hat{y}_{u,v})
\end{aligned} \tag{6}
$$

where $y_{u,v} = 1$ if user u previously interacted with item v, otherwise $y_{u,v} = 0$.

Generally, the recommendation model g can be any ranking model as long as it takes the user-feature and the item-feature vectors as the input. The implementation and training of g will be detailed in the experiment section.

Finally, given $\{\mathcal{U}, \mathcal{V}, \mathcal{T}, \boldsymbol{A}, \boldsymbol{B}, g\}$, our task is to generate feature-based explanations in terms of recommendation disparity for the black-box recommendation model g. Besides, most of the important symbols used in the paper can be referred in Tab. 1.

Symbol	Description
$\mathcal{U}$	The set of users in a recommender system
$\mathcal{V}$	The set of items in a recommender system
$\mathcal{T}$	The set of user-item interaction in a recommender system
$\mathcal{F}$	The set of features in a recommender system
$\mathcal{S}$	The set of sentiments in a recommender system
m	The number of users
n	The number of items
r	The number of features
u	A user id in a recommender system
v	An item id in a recommender system
f	A feature index in a recommender system
s	A sentiment index in a recommender system
$\boldsymbol{A}$	A user-feature attention matrix
$\boldsymbol{B}$	A item-feature quality matrix
$\boldsymbol{A}^{cf}$	The user-feature attention matrix after intervention with Δ_u
$\boldsymbol{B}^{cf}$	The item-feature quality matrix after intervention with Δ_v
$\mathcal{G}_0$	The set of popular items
$\mathcal{G}_1$	The set of long-tailed items
y_{uv}	Ground-truth value of the pair (u, v)
$\hat{y}_{uv}$	Predicted value of the pair (u, v)
K	The length of the recommendation list
$\mathcal{R}_K$	The set of recommendation lists with length K for all users
Θ	Parameters of black-box recommendation model

Table 1: Summary of the notations in this work.

3.3 Fairness and Disparity

In this work, we consider explaining the popularity bias in recommendation. Given a recommendation model g, we will have a certain recommendation result $\mathcal{R}_K = \{\mathcal{R}(u_1, K), \mathcal{R}(u_2, K), \cdots, \mathcal{R}(u_m, K)\}$ containing all users' top-K recommendation lists. These recommendations determine the exposures of items, which is used to measure the fairness and disparity of the model. We then split items into two groups based on their number of exposures in the recommendation list and denote $\mathcal{G}_0$ as popular item group and $\mathcal{G}_1$ as long-tailed item group. Based on the above notations, we list some popular algorithmic fairness definitions related to popularity bias as follows:

3.3.1 *Demographic Parity (DP).*

Demographic parity in recommendation scenarios requires that the average exposure of the items from each group is equal [19, 40]. First, given $\mathcal{R}_K$, we denote the number of exposure in group $\mathcal{G}_l$ as

$$
\text{Exposure } (\mathcal{G}_l | \mathcal{R}_K) = \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{R}(u,K)} \mathcal{I}(v \in \mathcal{G}_l), l \in \{0, 1\}. \tag{7}
$$

where $\mathcal{I}$ is the indicator function.

Then, we can express demographic parity fairness as follows,

$$
\frac{\text{Exposure } (\mathcal{G}_0 | \mathcal{R}_K)}{|\mathcal{G}_0|} = \frac{\text{Exposure } (\mathcal{G}_1 | \mathcal{R}_K)}{|\mathcal{G}_1|}, \tag{8}
$$

where groups $\mathcal{G}_0$ and $\mathcal{G}_1$ are divided based on the item popularity in the recommendation scenario.

3.3.2 *Exact-K Fairness (EK).*

Following [19], we can also use the Exact-K fairness in ranking, which requires the proportion/chance of protected candidates in every recommendation list with length K remains statistically indistinguishable from a given maximum α. This kind of fairness constraint is more suitable and feasible in practice for recommender systems as the system can adjust the value of α. The concrete form of this fairness is shown as below,

$$
\frac{\text{Exposure } (\mathcal{G}_0 | \mathcal{R}_K)}{\text{Exposure } (\mathcal{G}_1 | \mathcal{R}_K)} = \alpha \tag{9}
$$

where $\alpha \in (0, 1)$. Note that when $\alpha = \frac{|\mathcal{G}_0|}{|\mathcal{G}_1|}$ and the equation holds strictly, the above expression would be exactly the same as demographic parity.

3.3.3 *Disparity.*

In practice, we can take the difference between the two sides of the equalities in the above definitions as a quantification measure for disparity. For example,

$$
\Psi_{DP} = |\mathcal{G}_1| \cdot \text{Exposure } (\mathcal{G}_0 | \mathcal{R}_K) - |\mathcal{G}_0| \cdot \text{Exposure } (\mathcal{G}_1 | \mathcal{R}_K) \tag{10}
$$

$$
\Psi_{EK} = \text{Exposure } (\mathcal{G}_0 | \mathcal{R}_K) - \alpha \cdot \text{Exposure } (\mathcal{G}_1 | \mathcal{R}_K) \tag{11}
$$

are two popular algorithm disparity measures used in fairness learning algorithms [19].

3.4 Counterfactual Reasoning

With the above notations and definitions of item exposure fairness, we can measure the disparity of the top-K recommendation result $\mathcal{R}_K$. Then, the objective of our counterfactual reasoning problem is to generate feature-based explanations for the given black-box recommendation model g. The essential idea of the proposed explanation model is to discover a slight change Δ_v on each feature via solving a counterfactual optimization problem, which minimizes the disparity and a perturbation constraint that represents the effort to change the disparity.

Specially, for each user-feature vector $\boldsymbol{A}_{:f}$, we slightly intervene with a vector $\Delta_u \in \mathrm{R}^m$ (and for each item-feature vector $\boldsymbol{B}_{:f}$, we intervene with $\Delta_v \in \mathrm{R}^n$), more specifically, the value of certain user feature f for all users $\boldsymbol{A}_{:f}$ will be added to Δ_u and get A^{cf}, (or the value of certain item feature f for all items $\boldsymbol{B}_{:f}$ will be added to Δ_v and get $\boldsymbol{B}^{cf}$). With the new user-feature matrix $\boldsymbol{A}^{cf}$ and item-feature matrix $\boldsymbol{B}^{cf}$, g will change the recommendation result from $\mathcal{R}_K$ to a counterfactual result $\mathcal{R}_K^{cf}$. More importantly, this will also change the fairness measure of that result to Ψ^{cf}, where Ψ^{cf} can either be Ψ_{EK}^{cf} or Ψ_{DP}^{cf} depending on the choice of disparity. And our goal is to look for the minimum intervention on user/item feature that is able to result in the greatest reduction in terms of disparity or (un)fairness. Thus, objective function would be:

$$\min \|\Psi^{cf}\|_2^2 + \lambda\|\Delta\|_2 \tag{12}$$

where Δ can be either Δ_u or Δ_v or the concatenation of them ($\Delta = [\Delta_u, \Delta_v]$), $\lambda \in (0, 1)$ is a hyper-parameter that is used to control the weight between the two terms, and Ψ^{cf} can be:

$$\begin{aligned}\Psi_{EK}^{cf} &= \text{Exposure}(\mathcal{G}_0|\mathcal{R}_K^{cf}) - \alpha \cdot \text{Exposure}(\mathcal{G}_1|\mathcal{R}_K^{cf}) \\ &= \sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{R}^{cf}(u,K)} \mathcal{I}(v \in \mathcal{G}_0) - \alpha \cdot \sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{R}^{cf}(u,K)} \mathcal{I}(v \in \mathcal{G}_1)\end{aligned} \tag{13}$$

$$\begin{aligned}\Psi_{DP}^{cf} &= \text{Exposure}(\mathcal{G}_0|\mathcal{R}_K^{cf}) - \frac{|\mathcal{G}_0|}{|\mathcal{G}_1|} \cdot \text{Exposure}(\mathcal{G}_1|\mathcal{R}_K^{cf}) \\ &= \sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{R}^{cf}(u,K)} \mathcal{I}(v \in \mathcal{G}_0) - \frac{|\mathcal{G}_0|}{|\mathcal{G}_1|} \cdot \sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{R}^{cf}(u,K)} \mathcal{I}(v \in \mathcal{G}_1)\end{aligned} \tag{14}$$

A major challenge to optimize Eq. (12) is the non-differentiable nature of Ψ^{cf}. As a relaxation, we replace the indicator function $\mathcal{I}(\cdot)$ in the original definition (Eq. (14) or Eq. (13)) with $g(\cdot, \cdot)$, which is the predicted ranking score, and normalize the final results to stabilize the gradients of the objective function. And the resulting disparity metric $\tilde{\Psi}^{cf}$ becomes:

$$\tilde{\Psi}_{EK}^{cf} = \frac{\sum_{u\in\mathcal{U}} \left(\sum_{v\in\mathcal{G}_0\cap\mathcal{R}^{cf}(u,K)} g(\boldsymbol{A}_u^{cf}, \boldsymbol{B}_v^{cf}) - \alpha \sum_{v\in\mathcal{G}_1\cap\mathcal{R}^{cf}(u,K)} g(\boldsymbol{A}_u^{cf}, \boldsymbol{B}_v^{cf})\right)}{\sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{R}^{cf}(u,K)} g(\boldsymbol{A}_u^{cf}, \boldsymbol{B}_v^{cf})} \tag{15}$$

or becomes:

$$\tilde{\Psi}_{DP}^{cf} = \frac{\sum_{u\in\mathcal{U}} \left(\sum_{v\in\mathcal{G}_0\cap\mathcal{R}^{cf}(u,K)} g(\boldsymbol{A}_u^{cf}, \boldsymbol{B}_v^{cf}) - \frac{|\mathcal{G}_0|}{|\mathcal{G}_1|} \sum_{v\in\mathcal{G}_1\cap\mathcal{R}^{cf}(u,K)} g(\boldsymbol{A}_u^{cf}, \boldsymbol{B}_v^{cf})\right)}{\sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{R}^{cf}(u,K)} g(\boldsymbol{A}_u^{cf}, \boldsymbol{B}_v^{cf})} \tag{16}$$

Thus, our final objective for a given feature is

$$\min \|\tilde{\Psi}^{cf}\|_2^2 + \lambda\|\Delta\|_2. \tag{17}$$

The first term aims to realize the greatest reduction in terms of pre-defined disparity or (un)fairness. The second perturbation constraint represents the edit distance between original inputs and the corresponding counterfactuals. Finally, for each feature, we solve a optimization problem defined as Eq. (17) and use the corresponding counterfactual recommendation result to calculate the explainability score, which will be detailed in the next section.

3.5 Generate Feature-based Explanations

For each feature in the feature space, we will solve the optimization problem defined as Eq. (17) and consider Δ as the only trainable parameters. Once finished optimizing, we will get the "minimial" change Δ and the corresponding recommendation results under such change to that feature. Then, we use *Proximity*—the average edit distance between original inputs and the corresponding counterfactuals—to measure the degree of perturbations. And we use *Validity*—the changes of fairness caused by the feature's perturbation—to measure the degree of influence on fairness [34, 35, 43].

$$\textit{Proximity} = \|\Delta\|_2^2 \tag{18}$$

$$\begin{aligned}\textit{Validity} = &\frac{\text{Exposure}\,(\mathcal{G}_0|\mathcal{R}_K) - \text{Exposure}\,(\mathcal{G}_1|\mathcal{R}_K)}{m \cdot K} \\ &- \frac{\text{Exposure}\left(\mathcal{G}_0|\mathcal{R}_K^{cf}\right) - \text{Exposure}\left(\mathcal{G}_1|\mathcal{R}_K^{cf}\right)}{m \cdot K},\end{aligned} \tag{19}$$

where m is the number of users and K is the length of recommendation lists.

Finally, the explainability score (*ES*) is the linear combination of *Proximity* and *Validity*, which is shown as follows:

$$ES = \textit{Validity} - \beta \cdot \textit{Proximity}, \tag{20}$$

where $\beta \in (0, 1)$ and larger score represents better explainability.

This score determines the ranking of a feature in terms of its ability to reduce the disparity of model g while keeping the perturbation small. Note that the original value of the feature correspond to the optimal recommendation utility of $\mathcal{R}_K$ that the model g learned, so larger proximity score may imply a greater sacrifice of utility. Thus, the inclusion of this term in the objective function and the scoring function will result in an explanation finding process that is aware of the influence on both the fairness and recommendation utility.

4 EXPERIMENTS

4.1 Datasets

To evaluate the models under different data scales, data sparsity and application scenarios, we perform experiments on three real-world datasets. Some basic statistics of the experimental datasets are shown in Table 2.

- **Yelp** The Yelp dataset[2] contains users' reviews on various kinds of businesses such as restaurants, dentists, salons, etc.
- **Amazon** The Amazon dataset [32] contains user reviews on products in Amazon e-commerce system[3]. The Amazon dataset contains 29 sub-datasets corresponding to 29 product categories. We adopt two datasets of different scales to evaluate our method, which are *CDs & Vinyl* and *Electronics*.

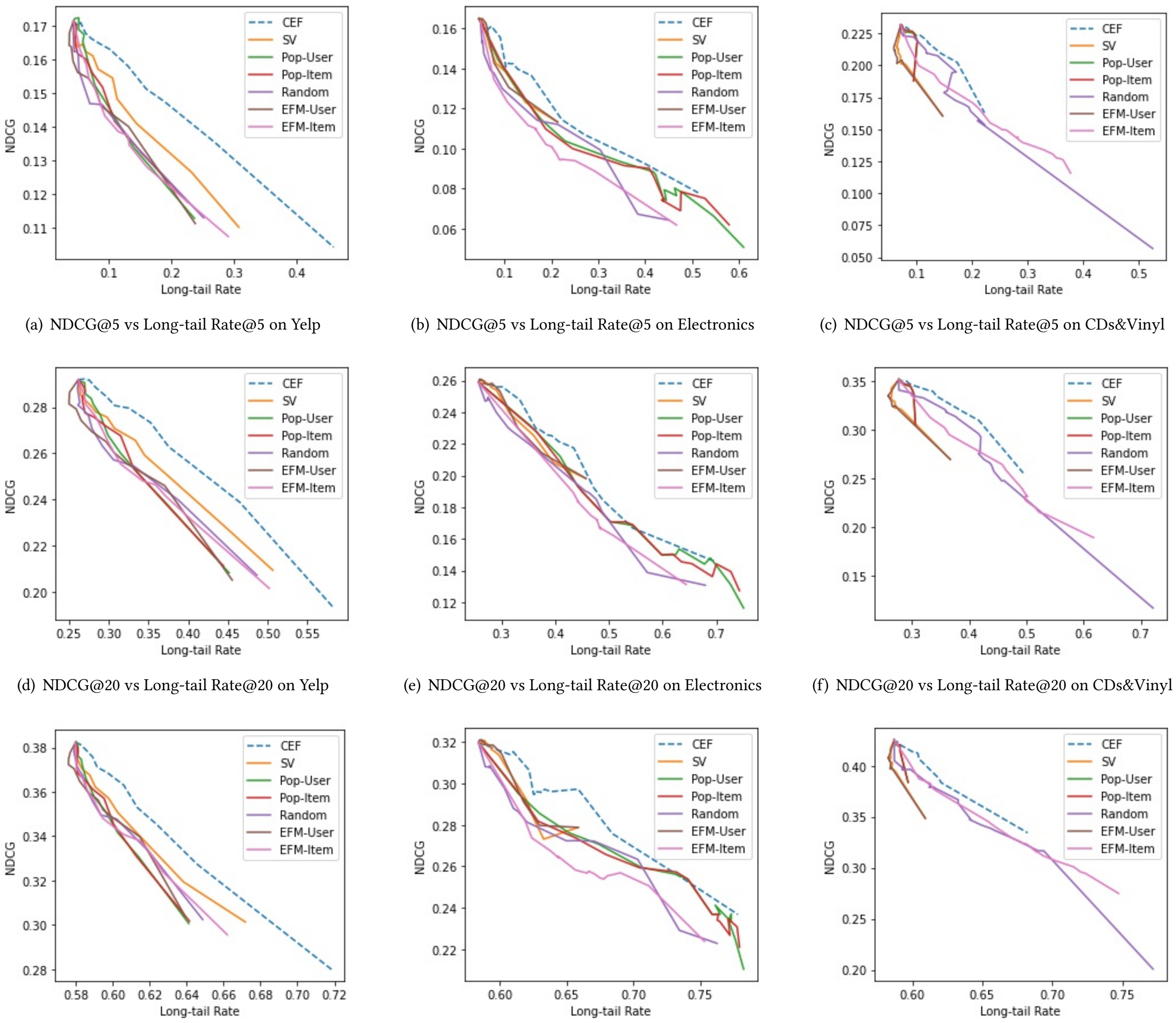
Since the Yelp and Amazon review datasets are very sparse, similar as previous work [41, 45, 55], we remove the users and items with fewer than 20 reviews. For each dataset, we first sort the records of each user based on the timestamp, and then hold-out the last 5 interacted items together with 100 randomly sampled negative items for each user to serve as the test data to evaluate black-box recommenders and do fairness explanation. The last item in the training set of each user is put into the validation set. Since we focus on item exposure fairness, we need to split items into two groups $\mathcal{G}_0$ and $\mathcal{G}_1$ based on item popularity. It would be desirable if we have the item impression/listing information and use it to group items, however, since Yelp and Amazon datasets are public dataset and only have interaction data, we use the number of interaction to group items in them. Specifically, for Yelp and Amazon review datasets, the top 20% items in terms of number of interactions belong to the popular group $\mathcal{G}_0$, and the remaining 80% belong to the long-tail group $\mathcal{G}_1$.

4.2 Black-box Recommender System

As mentioned before, we first follow prior works [11, 45, 54] to build a user-feature attention matrix and an item-feature quality

[2] https://www.yelp.com/dataset

[3] http://jmcauley.ucsd.edu/data/amazon/links.html



(a) NDCG@5 vs Long-tail Rate@5 on Yelp

(b) NDCG@5 vs Long-tail Rate@5 on Electronics

(c) NDCG@5 vs Long-tail Rate@5 on CDs&Vinyl

(d) NDCG@20 vs Long-tail Rate@20 on Yelp

(e) NDCG@20 vs Long-tail Rate@20 on Electronics

(f) NDCG@20 vs Long-tail Rate@20 on CDs&Vinyl

(g) NDCG@50 vs Long-tail Rate@50 on Yelp

(h) NDCG@50 vs Long-tail Rate@50 on Electronics

(i) NDCG@50 vs Long-tail Rate@50 on CDs&Vinyl

Figure 1: The accuracy-fairness trade-off curves for NDCG and Long-tail Rate on various datasets. The upper-right corner (high accuracy, low disparity) is preferred. Each data point is generated by cumulatively removing top 5 features in the explanation lists provided by explanation methods.

Table 2: Basic statistics of the experimental datasets.

Dataset	#User	#Item	#Review	#Aspect	Density
Yelp	12,028	20,181	502,158	106	0.208%
CDs & Vinyl	3,225	46,709	179,992	118	0.119%
Electronics	2,762	19,449	51,777	77	0.096%

matrix, and use both matrices together with the user-item interaction history to train a feature-aware recommendation model.

In this work, to demonstrate the idea of counterfactual explainable fairness, we use a simple deep neural network as the implementation of the recommendation model g, which includes one fusion layer followed by three fully connected layers with size {256, 64, 1}. The architecture of the fusion layer depends on how we are going to merge the user-feature and item-feature vectors (as is provided in Eq. (4) and Eq. (3.2)). Specifically, for Element-wise Product merge, the fusion layer is {2*feature size, 256}, while for Concatenation merge, it is {feature size, 256}. The final output layer is a sigmoid activation function so as to map $\hat{y}_{u,v}$ into the range of $(0, 1)$.

Table 3: **Summary of recommendation performance on three datasets for black-box recommendation models using Element-wise Product merge (Eq.** (4)**) and Concatenation merge (Eq.** (3.2)**) in term of F1 and NDCG.**

Recommender	F1 (%)		NDCG (%)	
	@5 ↑	@20 ↑	@5 ↑	@20 ↑
Yelp				
Element-wise	17.161	16.563	16.069	29.192
Concatenation	16.266	16.929	17.338	29.780
Electronics				
Element-wise	15.112	13.975	16.384	25.886
Concatenation	15.083	14.044	16.350	25.946
CDs & Vinyl				
Element-wise	21.463	18.517	23.150	35.162
Concatenation	20.737	18.443	22.393	34.672

The model parameters are optimized by stochastic gradient descent (SGD) optimizer with a learning rate of 0.01. After the recommendation model is trained, all the parameters will be fixed in the counterfactual reasoning phase and explanation evaluation phase. The recommendation performance on Element-wise product merge (Eq. (4)) and Concatenation merge (Eq. (3.2)) are presented in Tab. 3. The evaluations of fairness explanation methods presented in the experiment section are based on Element-wise product merge (Eq. (4)), while those based on Concatenation merge (Eq. (3.2)) are provided in the Appendix.

4.3 Baselines

Since there is no existing method specifically designed to explain disparity in recommendation. We adopt the following explanation methods as baselines:

- **Random**: We randomly choose multiple features from the feature space without replacement and use them as explanation results.
- **Popularity**: We rank all the features in the user-feature matrix and item feature matrix based on their number of existences, and select the top ones as explanations, and denote them as **Pop-User** and **Pop-Item**, respectively.
- **EFM** [55]: The Explicit Factor Model (EFM) for explainable recommendation. This work combines matrix factorization with sentiment analysis technique to align latent factors with explicit aspects. In this way, it predicts the user-feature preference scores and item-feature quality scores. The orgianl EFM uses the element-wise product of user-feature vector and item-feature vector and select the top ones as explanations to a given user-item pair. To generate global explanations, we calculate the average value of each feature from both user side and item side, and use them as explanations. Therefore, we have **EFM-User** and **EFM-Item**.
- **Feature-based Explanation by Shapley Values (SV)**: Begley et al. [4] leveraged Shapley value based methods to attribute the model disparity as the sum of individual contributions from input features to understand which feature contributes more or less to the model disparity. Considering the large number of features in the feature space, instead of using all possible coalitions, which is $r!$, we randomly sample 100 feature coalitions to calculate the Shapley value for each feature.

For CEF, we choose to minimize Eq. (17), where $\Delta = [\Delta_u, \Delta_v]$, $\Psi^{cf} = \Psi^{cf}_{DP}$ (Eq. (14)), and $\frac{|\mathcal{G}_0|}{|\mathcal{G}_1|} = \frac{1}{4}$. We set the hyper-parameter $\lambda = 1$ and $K = 5$. The model parameters are optimized by Adam optimizer with a learning rate of 0.01.

4.4 Evaluation Methods and Metrics

Once we obtained the feature-based explanations from each baseline as well as our proposed CEF, we need to compare the effectiveness of these results, in other words, their contributions to the fairness-utility trade-off. In order to evaluate the feature-based explanations, we follow the widely deployed erasure-based evaluation criterion in explainable AI. The intuition behind the erasure-based criterion is to measure how much the model performance would drop after the set of the "most important" features in an explanation is removed [50, 52]. Similarly, in the setting of explainable fairness, we use it to measure the fairness-utility trade-off in recommendation, namely, how much the recommendation performance would drop and how much the recommendation fairness would improve after the set of the "most important" features in an explanation is removed. Specifically, for each feature-based explanation result, we erase the set of the "most important" features in both the user-feature and item-feature matrices for all users and items, then input the erased user-feature and item-feature matrices into pre-trained recommendation model g to generated a new recommendation results. Based on the recommendation performance and fairness of the new results, we compare the effectiveness of each explainable fairness methods.

We select several most commonly used top-K ranking metrics to evaluate the model's recommendation performance after erasure, including **F1 Score**, and **NDCG**. For fairness evaluation, we define **Long-tail Rate**, which simply refers to the ratio of the number of long-tailed items in the recommendation list to the total number of items in the list. We also employ **KL-divergence** (KL) to compute the expectation of the difference between protected group membership at top-K vs. in the over-all population, which is shown as below,

$$d_{KL}(D_1 || D_2) = \sum_j D_1(j) \ln \frac{D_1(j)}{D_2(j)} \tag{21}$$

where D_1 represents the true group distribution between $\mathcal{G}_0$ and $\mathcal{G}_1$ in top-K recommendation list, and $D_2 = [\frac{|\mathcal{G}_0|}{|\mathcal{V}|}, \frac{|\mathcal{G}_1|}{|\mathcal{V}|}]$ represents their ideal distribution of the overall population.

4.5 Experimental Results

The major experimental results are shown in Fig. 1, where we plot the fairness-utility trade-off, i.e., the relationship between NDCG and Long-tail Rate (namely, 1-Popularity Rate) with different length of recommendation lists (@5, @20, @50). Since the relationship between F1 and and Long-tail Rate has very similar conclusions, we choose not to present them here. Each data point Fig. 1 is generated by cumulatively removing top 5 features in the remaining explanation list provided by each explanation method. We also present the

Table 4: Summary of the performance and fairness on three datasets. We evaluate for ranking ($F1$ and $NDCG$, in percentage (%) values, % symbol is omitted in the table for clarity) and fairness ($KL\ Divergence$ and $Long - tail\ Rate$, also in % values) with top 5 recommended items, and E is the number of erased features. Bold scores are used to indicate the greatest values.

Methods	F1@5(%) ↑			NDCG@5 (%) ↑			Long-tail Rate@5 (%) ↑			KL@5 (%) ↓		
	E=5	E=10	E=20	E=5	E=10	E=20	E=5	E=10	E=20	E=5	E=10	E=20
Yelp												
Random	15.671	15.345	14.809	16.788	16.255	15.674	4.3191	4.8066	5.2302	10.506	9.6994	9.0410
Pop-User	**16.074**	**15.636**	14.956	17.236	**16.748**	15.983	4.6047	6.0428	6.6289	10.026	7.8770	7.1107
Pop-Item	16.050	15.498	14.868	**17.055**	16.522	16.013	4.7180	4.5703	6.3580	9.8421	10.083	7.4577
EFM-User	15.735	15.370	14.710	16.804	16.392	15.626	3.7084	3.9940	5.0086	11.598	11.075	9.3808
EFM-Item	15.538	14.434	13.533	16.558	15.406	14.320	**5.0874**	6.8406	9.3622	**9.2588**	6.8477	4.2092
SV	15.680	15.188	14.814	16.700	16.235	15.719	4.4570	6.3974	8.2688	10.272	7.4064	5.2486
CEF	15.897	15.513	**15.296**	17.015	16.635	**16.309**	5.0233	**7.1706**	**10.169**	9.3579	**6.4518**	**3.5328**
Electronics												
Random	14.981	14.960	14.945	15.253	15.272	15.336	5.2715	5.4018	5.3439	8.9788	8.7846	8.8705
Pop-User	13.330	11.940	9.9782	14.417	12.795	10.361	8.6676	13.164	22.947	4.8522	1.6141	**0.2621**
Pop-Item	13.149	11.701	9.6017	14.118	12.482	9.9908	**9.6886**	**14.460**	**24.540**	**3.9269**	**1.0372**	0.6115
EFM-User	**15.018**	15.018	15.018	**16.454**	**16.451**	16.426	4.6125	4.4822	4.7139	10.014	10.230	9.8487
EFM-Item	12.541	11.622	10.586	13.453	12.314	10.976	7.7552	10.644	16.857	5.7903	3.1690	0.3218
SV	15.061	**15.126**	**15.112**	16.379	16.418	**16.487**	5.0615	4.9312	4.9674	9.2987	9.5019	9.4451
CEF	14.829	14.887	13.164	15.956	16.115	14.149	6.5821	7.1976	10.275	7.1697	6.4201	3.4500
CDs & Vinyl												
Random	**21.463**	21.246	21.103	22.131	21.968	21.821	7.2062	7.3612	7.5906	6.4102	6.2307	5.9715
Pop-User	21.413	21.432	**21.432**	23.118	23.133	**23.162**	7.1937	7.1999	7.2496	6.4247	6.4174	6.3596
Pop-Item	21.457	**21.469**	21.413	**23.156**	**23.196**	23.150	7.2062	7.2062	7.2186	6.4102	6.4102	6.3957
EFM-User	20.241	20.210	20.055	21.621	21.594	21.482	6.1395	6.0651	6.0093	7.7465	7.8468	7.9226
EFM-Item	19.968	18.381	17.159	21.653	19.972	18.619	**8.5271**	**10.449**	**14.325**	**4.9896**	**3.3154**	**1.0908**
SV	20.675	20.700	20.545	22.290	22.283	22.174	6.9271	6.8403	6.9147	6.7424	6.8481	6.7574
CEF	**21.463**	21.438	21.333	23.099	23.061	22.962	7.2124	7.2496	7.4046	6.4029	6.3596	6.1811

values of F1@5, NDCG@5, Long-tail Rate@5 and KL@5 in Tab. 4 after removing top-5, top-10, top-20 features in each explanation result to quantitatively analyse the results.

First, in Fig. 1 and Tab. 4, we can easily find that all the methods, even randomly selecting features and erasing them, can improve recommendation fairness. Besides, the higher the number of features we erase, the lower the disparity rate we can achieve. This is easy to understand as erasing features will mitigate the representation gap between popular items and long-tailed items, causing more under-represented items to be recommended. However, it also brings huge decline to the recommendation performance. For example, compared with the original recommendation performance on NDCG@5, the method with the worst trade-off behavior drops relatively 2.119 % on Yelp, 21.786 % on Electronics, and 7.072 % on CDs & Vinly when deleting top 5 features. Second, we can see that even though the idea of using popular features as explanations is very intuitive, their performance may even worse than randomly selecting, which indicates that compared with fairness, popular features either from user side or item side are more sensitive to recommendation performance, while randomly selecting guarantees low probabilities of choosing those scarce features, which in turn results in better trade-off. Third, the performances of SV are much worse than CEF as it only explain disparity alone, ignoring the inherent trade-off between fairness and utility. Finally, in Fig. 1, where the blue dotted line represents the performance of our proposed CEF framework, it is obvious that the feature-based explanations provided by CEF are capable of achieving much better fairness-utility trade-off on datasets with various scales and densities. Specifically, compared with the original recommendation performance on NDCG@5, CEF method drops only relatively 0.851 % on Yelp, 2.682 % on Electronics, and 0.594 % on CDs & Vinly, while it increases relatively 13.431 % on Yelp, 25.73 % on Electronics, and on 3.085 % CDs & Vinly at Long-tail Rate@5.

4.6 Ablation

As mentioned in Sec. 3.4, the choice of Δ in the objective function (Eq. (12) or Eq. (17)) can either be Δ_u or Δ_v or both of them, depending on how we are going to intervene the given feature in the feature space. Besides, all the experimental results in Table 4 and Fig. 1 are based on intervening the given feature using both Δ_u and Δ_v (namely, $\Delta = [\Delta_u, \Delta_v]$). Therefore, to study how the choice of Δ is going to influence the experimental results, we run additional experiments based on the variants of the original CEF by either choosing Δ_u or Δ_v alone, denoted them as CEF-User and CEF-Item, respectively. The objective of CEF-User is $\min ||\tilde{\Psi}^{cf}_{EK}||_2^2 + \lambda||\Delta_u||_2$. And that of CEF-Item is $\min ||\tilde{\Psi}^{cf}_{EK}||_2^2 + \lambda||\Delta_v||_2$. For convenience,

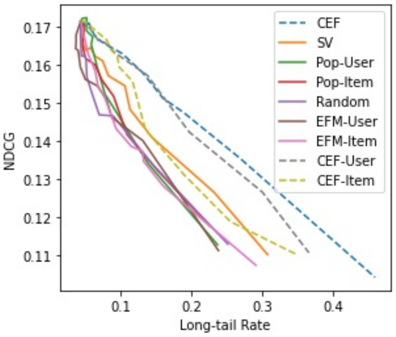


Figure 2: Ablation study on Yelp dataset.

we only present the results in Yelp dataset, as is shown in Fig. 3. The similar conclusions can also be proved using other datasets.

As is shown in Fig. 3, the evaluations of CEF-User and CEF-Item achieve worse fairness-utility trade-off when compared with the original CEF. This is understandable as CEF uses both Δ_u and Δ_v as its parameters, which is a much larger parameter space and can achieve better representations. Moreover, even though CEF-User and CEF-Item are worse than CEF, they are still far more better than most of the baselines, especially, CEF-User is better than all the baselines, which indicates the effectiveness of our proposed framework.

5 CONCLUSION AND FUTURE WORK

In this paper, we study the problem of explainable fairness in recommendation and propose a framework based on counterfactual reasoning, called CEF. To the best of our knowledge, this is the first time that the idea of explainable fairness has been introduced into the area of recommender systems. We design a learning-based counterfactual reasoning method to discover critical features that will significantly influence the fairness-utility trade-off and use them as fairness explanations for black-box feature-aware recommendation systems. Extensive experiments have been conducted to evaluate the effectiveness of our proposed frameworks and these explanations generated by CEF can achieve better fairness-utility trade-off than all the baselines when using them to do fair learning. In the future, we hope to design algorithmic methods that can generate multiple explanations at the same time without greedy choosing them through explainability scores.

A APPENDIX: BLACK-BOX RECOMMENDER SYSTEM BASED ON CONCATENATION MERGE

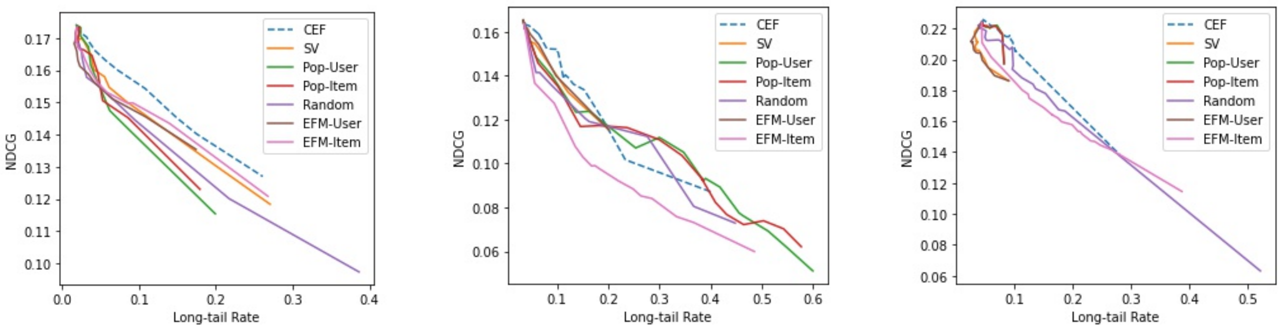
The experimental results on three datasets for black-box recommendation models using Concatenation merge (Eq. (3.2)) are shown in Fig. 3, where we plot the fairness-utility trade-off with the length of recommendation lists equals to 5. In Fig. 3, we can still find out that the feature-based explanations provided by CEF are capable of achieving much better fairness-utility trade-off on datasets with various scales and densities when used for fair learning, which further proves the effectiveness and generality of the proposed framework.

B CASE STUDY

In this section, we provide the top-5 feature-based explanations that are generated by each method on Yelp dataset. The explanation results are shown in Tab. 5, which exactly verifies our motivation that it's difficult to manually identify feature explanations for popularity bias in recommender system. For example, it is hard to tell how input features (like chicken, cheese, pizza) would influence the popularity bias in restaurant recommendation. Thus, we do need explainable fairness methods to identify such features in recommendation.

Method	Feature-based Explanations
Pop-User	food, service, chicken, prices, hour
Pop-Item	food, service, prices, visit, hour
EFM-User	store, patio, dishes, dish, rice
EFM-Item	flavor, decor, dishes, inside, cheese
SV	server, size, pizza, food, restaurant
CEF	meal, cheese, dish, chicken, taste

Table 5: Top-5 feature-based explanations on Yelp dataset.



(a) NDCG@5 vs Long-tail Rate@5 on Yelp

(b) NDCG@5 vs Long-tail Rate@5 on Electronics

(c) NDCG@5 vs Long-tail Rate@5 on CDs&Vinyl

Figure 3: The accuracy-fairness trade-off curves for NDCG@5 and Long-tail Rate@5 on three datasets using model with Concatenation merge. The upper-right corner (high accuracy, low disparity) is preferred. Each data point is generated by cumulatively removing top 5 features in the explanation lists provided by explanation methods.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 42–46.

[2] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286* (2019).

[3] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation. *Algorithms* (2018).

[4] Tom Begley, Tobias Schwedes, Christopher Frye, and Ilya Feige. 2020. Explainability for fair machine learning. *arXiv preprint arXiv:2010.07389* (2020).

[5] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD*.

[6] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research)*, Sorelle A. Friedler and Christo Wilson (Eds.), Vol. 81. PMLR, New York, NY, USA, 202–214.

[7] L Elisa Celis, Sayash Kapoor, Farnood Salehi, and Nisheeth Vishnoi. 2019. Controlling polarization in personalization: An algorithmic framework. In *Proceedings of the conference on fairness, accountability, and transparency*. 160–169.

[8] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*.

[9] Hanxiong Chen, Li Yunqi, Shi Shaoyun, Shuchang Liu, He Zhu, and Yongfeng Zhang. 2022. Graph Logic Reasoning for Recommendation and Link Prediction. In *Proceedings of the 15th WSDM*.

[10] Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. [n.d.]. Investigating the Impact of Gender on Rank in Resume Search Engines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*.

[11] Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 891–900.

[12] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to rank features for recommendation over multiple categories. In *SIGIR*.

[13] Xu Chen, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Visually explainable recommendation. *arXiv* (2018).

[14] Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. 2020. Fairness-aware explainable recommendation over knowledge graphs. In *Proceedings of the 43rd SIGIR*. 69–78.

[15] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable recommendation through attentive multi-view learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3622–3629.

[16] Ruoyuan Gao, Yingqiang Ge, and Chirag Shah. 2021. FAIR: Fairness-Aware Information Retrieval Evaluation. *arXiv preprint arXiv:2106.08527* (2021).

[17] Ruoyuan Gao and Chirag Shah. 2019. How Fair Can We Go: Detecting the Boundaries of Fairness Optimization in Information Retrieval. In *Proceedings of ICTIR '19*. ACM, New York, NY, USA, 229–236.

[18] Ruoyuan Gao and Chirag Shah. 2021. Addressing Bias and Fairness in Search Systems. In *Proceedings of the 44th International ACM SIGIR (SIGIR '21)*. 4. https://doi.org/10.1145/3404835.3462807

[19] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, et al. 2021. Towards Long-term Fairness in Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 445–453.

[20] Yingqiang Ge, Shuyuan Xu, Shuchang Liu, Zuohui Fu, Fei Sun, and Yongfeng Zhang. 2020. Learning Personalized Risk Preferences for Recommendation. In *Proceedings of the 43rd SIGIR*. 409–418.

[21] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *Proceedings of KDD*. ACM, 2221–2231.

[22] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. *WSDM* (2020).

[23] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *CIKM*.

[24] Mohammad Mahdi Kamani, Rana Forsati, James Z Wang, and Mehrdad Mahdavi. 2021. Pareto Efficient Fairness in Supervised Learning: From Extraction to Tracing. *arXiv preprint arXiv:2104.01634* (2021).

[25] Michael Kearns and Aaron Roth. 2019. *The ethical algorithm: The science of socially aware algorithm design*. Oxford University Press.

[26] Trung-Hoang Le and Hady W Lauw. 2021. Explainable Recommendation with Comparative Constraints on Product Aspects. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 967–975.

[27] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *Proceedings of the Web Conference 2021*. 624–632.

[28] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards Personalized Fairness Based on Causal Notion. In *Proceedings of the 44th SIGIR (SIGIR '21)*. 10. https://doi.org/10.1145/3404835.3462966

[29] Yunqi Li, Yingqiang Ge, and Yongfeng Zhang. 2021. Tutorial on Fairness of Machine Learning in Recommender Systems. In *Proceedings of the 30th CIKM*.

[30] Zachary Lipton, Julian McAuley, and Alexandra Chouldechova. 2018. Does mitigating ML's impact disparity require treatment disparity?. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[31] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[32] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[33] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a Fair Marketplace: Counterfactual Evaluation of the Trade-off Between Relevance, Fairness & Satisfaction in Recommendation Systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*.

[34] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. 2020. Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explorations Newsletter* 22, 1 (2020), 18–33.

[35] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.

[36] Weishen Pan, Sen Cui, Jiang Bian, Changshui Zhang, and Fei Wang. 2021. Explaining algorithmic fairness through fairness-aware causal path decomposition. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1287–1297.

[37] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[38] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *RecSys*.

[39] Lloyd S Shapley. 2016. *17. A value for n-person games*. Princeton University Press.

[40] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD*.

[41] 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*. 1784–1793.

[42] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual Explanations for Neural Recommenders. *SIGIR* (2021), 1627–1631.

[43] Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual Explanations for Machine Learning: A Review. *arXiv preprint arXiv:2010.10596* (2020).

[44] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*.

[45] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 165–174.

[46] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable Reasoning over Knowledge Graphs for Recommendation. *AAAI* (2019).

[47] Shuyuan Xu, Yunqi Li, Shuchang Liu, Zuohui Fu, Yingqiang Ge, Xu Chen, and Yongfeng Zhang. 2021. Learning Causal Explanations for Recommendation. *The 1st International Workshop on Causality in Search and Recommendation* (2021).

[48] Tao Yang and Qingyao Ai. 2021. Maximizing Marginal Fairness for Dynamic Learning to Rank. In *Proceedings of the Web Conference 2021*. 137–145.

[49] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *Advances in Neural Information Processing Systems*.

[50] Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking Cooperative Rationalization: Introspective Extraction and Complement Control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 4085–4094.

[51] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, Krishna P. Gummadi, and Adrian Weller. 2017. From Parity to Preference-Based Notions of Fairness in Classification. In *Proceedings of NIPS'17*.

[52] Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*. 260–267.

[53] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* (2020).

[54] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. *SIGIR* (2014), 83–92.

[55] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 83–92.

[56] Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2014. Do Users Rate or Review? Boost Phrase-Level Sentiment Labeling with Review-Level Sentiment Classification. In *SIGIR*. 1027–1030.