Counterfactual Explanation for Fairness in Recommendation

XIANGMENG WANG*, Data Science and Machine Intelligence Lab, University of Technology Sydney, Australia

QIAN LI*[†], School of Electrical Engineering Computing and Mathematical Sciences, Curtin University, Australia

DIANER YU, Data Science and Machine Intelligence Lab, University of Technology Sydney, Australia

QING LI, Hong Kong Polytechnic University, Hong Kong

GUANDONG XU[‡], Data Science and Machine Intelligence Lab, University of Technology Sydney, Australia

Fairness-aware recommendation eliminates discrimination issues to build trustworthy recommendation systems. Explaining the causes of unfair recommendations is critical, as it promotes fairness diagnostics, and thus secures users' trust in recommendation models. Existing fairness explanation methods suffer high computation burdens due to the large-scale search space and the greedy nature of the explanation search process. Besides, they perform score-based optimizations with continuous values, which are not applicable to discrete attributes such as gender and race. In this work, we adopt the novel paradigm of counterfactual explanation from causal inference to explore how minimal alterations in explanations change model fairness, to abandon the greedy search for explanations. We use real-world attributes from Heterogeneous Information Networks (HINs) to empower counterfactual reasoning on discrete attributes. We propose a novel *Counterfactual Explanation for Fairness (CFairER)* that generates attribute-level counterfactual explanations from HINs for recommendation fairness. Our *CFairER* conducts off-policy reinforcement learning to seek high-quality counterfactual explanations, with an attentive action pruning reducing the search space of candidate counterfactuals. The counterfactual explanations help to provide rational and proximate explanations for model fairness, while the attentive action pruning narrows the search space of attributes. Extensive experiments demonstrate our proposed model can generate faithful explanations while maintaining favorable recommendation performance. We release our code at https://anonymous.4open.science/r/CFairER-anony/.

CCS Concepts: • **Computing methodologies → Causal reasoning and diagnostics**; **Reinforcement learning**; • **Information systems → Personalization**.

Additional Key Words and Phrases: Explainable Recommendation; Fairness; Counterfactual Explanation; Reinforcement Learning

ACM Reference Format:
Xiangmeng Wang, Qian Li, Dianer Yu, Qing Li, and Guandong Xu. 2018. Counterfactual Explanation for Fairness in Recommendation. *ACM Trans. Inf. Syst.* 1, 1 (June 2018), 24 pages. https://doi.org/XXXXXXX.XXXXXXX

*Equal contribution.
[†]Corresponding author: qli@curtin.edu.au
[‡]Corresponding author: guandong.xu@uts.edu.au

Authors' addresses: Xiangmeng Wang, xiangmeng.wang@student.uts.edu.au, Data Science and Machine Intelligence Lab, University of Technology Sydney, Sydney, Australia; Qian Li, qli@curtin.edu.au, School of Electrical Engineering Computing and Mathematical Sciences, Curtin University, Perth, Australia; Dianer Yu, Dianer.Yu-1@student.uts.edu.au, Data Science and Machine Intelligence Lab, University of Technology Sydney, Sydney, Australia; Qing Li, qing-prof.li@polyu.edu.hk, Hong Kong Polytechnic University, Hong Kong; Guandong Xu, Guandong.Xu@uts.edu.au, Data Science and Machine Intelligence Lab, University of Technology Sydney, Sydney, Australia.

1 INTRODUCTION

Recommendation system (RS) as an information filtering tool has been a core in online services, e.g., e-commerce [9, 22]. It helps users discover their preferred items and benefit content providers to profit from item exposures. Despite the huge benefits, fairness issues refer to unfair allocations (i.e., exposures) of recommended items [28], caused by, e.g., gender discrimination, have attracted increasing attention in RS. Fairness-aware recommendation [13] has emerged as a promising solution to prevent unintended discrimination and unfairness in RS. It aims to find feasible algorithmic approaches that reduce the fairness disparity of recommendation results. Explaining why fairness disparity appears, i.e., *what causes unfair recommendation results*, would enhance the design of fairness-aware recommendation approaches by promoting model transparency and tracking unfair factors.

There are a few fairness explanation studies in the literature, which are mainly categorized as feature-based and aspect-based methods. Feature-based methods estimate the contribution scores of numerical features that impact model fairness. For instance, Begley et al. [2] explore fairness explanations based on Shapley value estimation for the classification task. They calculate Shapley values of every input features to reflect their significance and then generate explanations based on calculated values. However, this method is not applicable for deep recommendation models (e.g., neural networks [7, 23]), as the high complexity of Shapley value estimation becomes the major burden when input features are in high dimension and sparse. Another branch of aspect-based methods mainly perturbs user/item aspect scores and optimizes an explanation model to find perturbed aspects that affect the model fairness as explanations. For example, Ge et al. [16] perturb aspect scores within pre-defined user-aspect and item-aspect matrices and feed the perturbed matrices into a recommendation model. Those perturbed aspects that alter the fairness disparity of the recommendation model are considered aspect-based explanations. However, the perturbation space grows exponentially as the number of aspects increases, resulting in a large-scale search space to seek explanations.

The above fairness explanation methods suffer below issues: 1) These feature/aspect-based methods usually incur high computational costs due to the high dimensionality of search space and ultimately result in sub-optimal explanations. Besides, these methods are presented with the greedy nature of the explanation search process. They optimize explanation models using greedy feature/aspect scores as significance criteria and select top features/aspects as explanations, which might have the risk of introducing pseudo-explanations. 2) These score-based optimizations can only deal with continuous attributes and thus are not well-suited for handling discrete attributes. For example, assigning a continuous value, such as *gender*=0.19, to the discrete *gender* attribute is impractical in constructing explanations and provides no valuable clue to improve the explanation. Worse still, discrete attributes are frequently used in real-world recommendation models, as user and item profiles for training models are often generated through data tagging [20] on discrete attributes. For instance, movie recommendations [19, 34, 61] usually rely on movies tagged with discrete attributes such as genre, language, and release location. Consequently, score-based optimizations have limited capability in handling discrete attributes that are frequently encountered in recommendation scenarios.

Unlike previous works, we resort to counterfactual explanations [43] derived from causal inference to tackle the above issues. Counterfactual explanations address the fundamental question: *what the model fairness would be if a minimal set of factors (e.g., user/item features) had been different* [43]. In other words, they provide "what-if" explanations to determine the most vital and essential (i.e., *minimal*) factors that change model fairness. Unlike existing feature/aspect-based methods with greedy explanations, counterfactual explanations have the advantage of always being minimal w.r.t. the generated explanations and are faithful to model fairness changes. Moreover, we leverage real-world attributes from Heterogeneous Information Networks (HINs) [49], for counterfactual reasoning when dealing with discrete attributes. In contrast to value-based features and aspects, real-world attributes residing in HINs are presented as discrete nodes, with edges representing their connections. By utilizing attributes from HINs, we can overcome the limitation of score-based optimizations to directly measure whether the removal of specific attributes changes the model's fairness.

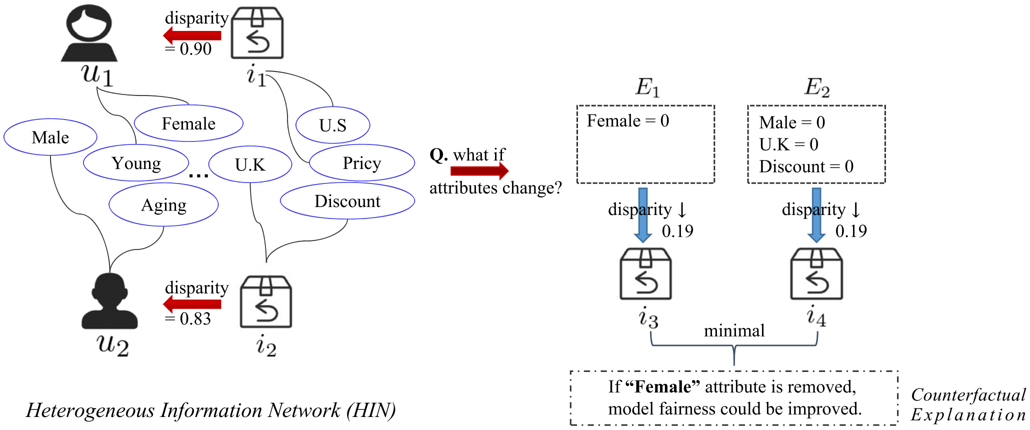


Fig. 1. Toy example of inferring the attribute-level counterfactual explanation for fairness.

Following the above intuition, we propose to generate attribute-level counterfactual explanations for fairness from a given HIN. We posit a novel definition of counterfactual explanation for fairness - *a minimal set of attributes from the HIN that changes model fairness disparity*. We use a toy example in Figure 1 to illustrate our idea. Given a recommendation i_1 for the user u_1 and an external HIN carrying their attributes, we want to know why i_1 causes discrimination in recommendation results. The counterfactual explanation performs "what-if" reasoning by altering the attributes of u_1 and i_1 and checking the fairness of the recommendation results. Both E_1 and E_2 are valid candidate explanations since they alter fairness disparities of recommendations (i.e., i_2, i_3) from 0.90 to 0.19. To determine which attributes are the primary reason for unfairness, the counterfactual explanation will uncover the minimal attribute changes, i.e., E_2, instead of utilizing attribute combinations in E_1. Thus, we could infer E_2 is the most vital reason for model unfairness. Besides, since a counterfactual explanation E_2 is minimal, it only reveals the essential attributes (i.e., "Female") that effectively explain unfairness, while discarding the irrelevant (i.e., pseudo) explanations, i.e., "U.S" and "Discount" in E_1.

We therefore propose a novel *Counterfactual Explanation for Fairness (CFairER)* within an off-policy reinforcement learning environment to find optimal attribute-level counterfactual explanations. Particularly, we focus on generating attribute-level counterfactual explanations for item exposure unfairness to promote the fair allocation of user-preferred but less exposed items. Note that the proposed approach is general and can be utilized in different recommendation scenarios that involve different fairness definitions. Specifically, we use a reinforcement learning agent in *CFairER* to optimize a fairness explanation policy by uniformly exploring candidate counterfactuals from a given HIN. We also devise attentive action pruning over the HIN to reduce the search space of reinforcement learning. Finally, our *CFairER* optimizes the explanation policy using an unbiased counterfactual risk minimization objective, resulting in accurate attribute-level counterfactual explanations for fairness. The contributions of this work are:

- We make the first attempt to leverage rich attributes in a Heterogeneous Information Network to offer attribute-level counterfactual explanations for recommendation fairness.

- We propose an off-policy learning framework to identify optimal counterfactual explanations, which is guided by an attentive action pruning to reduce the search space.
- We devise a counterfactual risk minimization for off-policy correction, so as to achieve unbiased policy optimization.
- Comprehensive experiments show the superiority of our method in generating trustworthy explanations for fairness while preserving satisfactory recommendation performance.

2 RELATED WORK

2.1 Fairness Explanation for Recommendation

Recommender systems have long dealt with major concerns of recommendation unfairness, which profoundly harm user satisfaction [13, 27] and stakeholder benefits [4, 17, 28]. Recent works on fairness-aware recommendation mainly discuss two primary topics, i.e., user-side fairness [8, 13, 27, 29, 57] and item-side fairness [1, 11, 14, 30]. User-side fairness concerns whether the recommendation is fair to different users/user groups, e.g., retaining equivalent accuracy or recommendation explainability. Relevant approaches attribute the causes of user-side unfairness to discrimination factors, such as sensitive features (e.g., gender [8, 57], race [29]) and user inactiveness [13, 27], etc. They mainly propose fairness metrics to constraint recommendation models (e.g., collaborative filtering [57]) to produce fair recommendations. For example, Yao et al. [57] study the unfairness of collaborative filtering (CF)-based recommenders on gender-imbalanced data. They propose four metrics to assess different types of fairness, then add these metrics as constraints to the CF model learning objective to produce fair recommendations. Li et al. [27] investigate the unfair recommendation between active and inactive user groups, and provide a re-ranking approach to mitigate the activity unfairness by adding constraints over evaluation metrics of ranking. As modern content providers are more concerned about user privacy, it is generally not easy to access sensitive user features for the recommendation [36]. Meanwhile, users often prefer not to disclose personal information that raises discrimination [3]. Thus, another topic of item-side fairness-aware recommendation [1, 11, 14, 30] is interested in examining whether the recommendation treats items fairly, e.g., similar ranking prediction errors for different items, fair allocations of exposure to each item. For instance, Abdollahpouri et al. [1] address item exposure unfairness in learning-to-rank (LTR) recommenders. They include a fairness regularization term in the LTR objective function, which controls the recommendations favored toward popular items. Ge et al. [14] consider the dynamic fairness of item exposure due to changing group labels of items. They calculate the item exposure unfairness with a fairness-related cost function. The cost function is merged into a Markov Decision Process to capture the dynamic item exposure for recommendations. Liu et al. [30] focus on item exposure unfairness in interactive recommender systems (IRS). They propose a reinforcement learning method to maintain a long-term balance between accuracy and exposure fairness in IRS.

Despite the great efforts, fairness-aware recommendations mitigate user and item unfairness in a black-box manner but do not explain why the unfairness appears. Understanding the "why" is desirable for both model transparency [28] and facilitates data curation to remove unfair factors [50]. Limited pioneering studies are conducted to explain fairness. Begley et al. [2] estimate Shapley values of input features to search which features contribute more to the model unfairness. Ge et al. [16] develop an explainable fairness model for recommendation to explain which item aspects influence item exposure fairness. They perform perturbations on item aspect scores, then apply perturbed aspect scores on two pre-defined matrices to observe fairness changes. These prior efforts suffer from major limitations: 1) The high computational burden caused by the large-scale search space and the greedy nature of the explanation search process.

2) They generate explanations by feature [2] or aspect [16] scores, which do not apply to discrete attributes such as gender and race. Our work conducts counterfactual reasoning to seek minimal sets of attributes as explanations. We also reduce the large search space by attentive action pruning in the off-policy learning environment. Meanwhile, we consider explaining recommendation unfairness based on attributes from a Heterogeneous Information Network, which is expected to be wildly applicable.

2.2 Heterogeneous Information Network in Recommendation

Heterogeneous Information Network (HIN) is a powerful structure that allows for the heterogeneity of its recorded data, i.e., various types of attributes, thus providing rich information to empower recommendations [46, 49]. HINs have been wildly adopted in recommendation models to boost performance; representative works cover context-based filtering (e.g., SemRec [38], HERec [37]) and knowledge-based systems (e.g., MCrec [24], HAN [45]). For instance, HERec [37] embeds meta-paths within a HIN as dense vectors, then fuses these HIN embeddings with user and item embeddings to augment the semantic information for recommendations. MCrec [24] leverages a deep neural network to model meta-path-based contextual embeddings and propagates the context to user and item representations with a co-attention mechanism. Those recommendation models observe promising improvements by augmenting contextual and semantic information given by HINs. Despite the great efforts, prior works do not consider using the HIN to explain unfair factors in recommendations. Novel to this work, we first attempt to leverage rich attributes in a HIN to provide counterfactual explanations for item exposure fairness.

2.3 Counterfactual Explanation

Counterfactual explanations have been considered as satisfactory explanations [26, 53] and elicit causal reasoning in humans [6, 58]. Works on counterfactual explanations have been proposed very recently to improve the explainability of recommendations. Xiong et al. [54] propose a constrained feature perturbation on item features and consider the perturbed item features as explanations for ranking results. Ghazimatin et al. [18] perform random walks over a Heterogeneous Information Network to look for minimal sets of user action edges (e.g., click) that change the PageRank scores. Tran et al. [41] identify minimal sets of user actions that update the parameters of neural models. Our work differs from prior works on counterfactual explanations by two key points: 1) In terms of problem definition, they generate counterfactual explanations to explain user behaviors (e.g., click [18, 41]) or recommendation (e.g., ranking [54]) results. Our method generates counterfactual explanations to explain which attributes affect recommendation fairness. 2) In terms of technique, our method formulates counterfactual reasoning as reinforcement learning, which can deal with ever-changing item exposure unfairness.

3 PRELIMINARY

We first introduce the Heterogeneous Information Network that offers real-world attributes for fairness explanation learning. We then give the key terminologies, including fairness disparity evaluation and counterfactual explanation for fairness.

3.1 Heterogeneous Information Network

Creating fairness explanations requires auxiliary attributes containing possible factors (e.g., user gender) that affect recommendation fairness (cf. Figure 1). Heterogeneous Information Network (HIN) has shown its power in modeling various types of attributes, e.g., user social relations, item brand. In particular, suppose we have the logged data that records users' historical behaviors (e.g., clicks) in the recommendation scenario. Let $\mathcal{U} \in \mathbb{R}^M$, $\mathcal{I} \in \mathbb{R}^N$ denote the sets of users and items, respectively. We can define a user-item interaction matrix $Y = \{y_{uv} \mid u \in \mathcal{U}, v \in \mathcal{I}\}$ according to the logged data. We also have additional attributes from external resources that profile users and items, e.g., users' genders, items' genres. The connections between all attributes and users/items are absorbed in the relation set $\mathcal{E}$. Those attributes, with their connections with user-item interactions, are uniformly formulated as a HIN. Formally, a HIN is defined as $\mathcal{G} = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}' = \mathcal{U} \cup \mathcal{I} \cup \mathcal{V}_U \cup \mathcal{V}_I$, and $\mathcal{E}' = \{\mathbb{I}(y_{uv})\} \cup \mathcal{E}$. $\mathbb{I}(\cdot)$ is an edge indicator that denotes the observed edge between user u and item v when $y_{uv} \in Y = 1$. $\mathcal{V}_U$ and $\mathcal{V}_I$ are attribute sets for users and items, respectively. Each node $n \in \mathcal{V}'$ and each edge $e \in \mathcal{E}'$ are mapped into specific types through node type mapping function: $\phi : \mathcal{V}' \to \mathcal{K}$ and edge type mapping function: $\psi : \mathcal{E}' \to \mathcal{J}$. $\mathcal{G}$ maintain heterogeneity, i.e., $|\mathcal{K}| + |\mathcal{J}| > 2$.

3.2 Fairness Disparity

We consider explaining the item exposure (un)fairness in recommendations. We first split items in historical user-item interactions into head-tailed (i.e., popular) group G_0 the long-tailed group G_1 [1]. Following previous works [14, 16], we use demographic parity (DP) and exact-K (EK) defined on item subgroups to measure whether a recommendation result is fair. In particular, DP requires that each item has the same likelihood of being classified into G_0 and G_1. EK regulates the item exposure across each subgroup to remain statistically indistinguishable from a given maximum α. By evaluating the deviation of recommendation results from the two fairness criteria, we can calculate the fairness disparity, i.e., to what extent the recommendation model is unfair. Formally, giving a recommendation result $H_{u,K}$, the fairness disparity $\Delta(H_{u,K})$ of $H_{u,K}$ is:

$$\begin{aligned} &\Delta(H_{u,K}) = |\Psi_{DP}| + \lambda \, |\Psi_{EK}| \, , \\ &\Psi_{DP} = |G_1| \cdot \text{ Exposure } \left(G_0 \mid H_{u,K}\right) - |G_0| \cdot \text{ Exposure } \left(G_1 \mid H_{u,K}\right), \\ &\Psi_{EK} = \alpha \cdot \text{ Exposure } \left(G_0 \mid H_{u,K}\right) - \text{ Exposure } \left(G_1 \mid H_{u,K}\right) \end{aligned} \tag{1}$$

where $\Delta(\cdot)$ is the fairness disparity metric that quantifies model fairness status. λ is the trade-off parameter between DP and EK. Exposure $\left(G_j \mid H_{u,K}\right)$ is the item exposure number of $H_{u,K}$ within G_j w.r.t. $j \in \{0, 1\}$.

3.3 Counterfactual Explanation for Fairness

This work aims to generate attribute-level counterfactual explanations for item exposure fairness. In particular, we aim to find the "minimal" changes in attributes that reduce the fairness disparity (cf. Eq. (1)) of item exposure. Formally, given historical user-item interaction $Y = \{y_{uv} \mid u \in \mathcal{U}, v \in \mathcal{I}\}$, and user attribute set $\mathcal{V}_U$ and item attribute set $\mathcal{V}_I$ extracted from an external Heterogeneous Information Network (HIN) $\mathcal{G} = (\mathcal{V}', \mathcal{E}')$. Suppose there exists a recommendation model that produces the recommendation result $H_{u,K}$ for user u. Given all user-item pairs (u, v) in $H_{u,K}$, our goal is to find a minimal attributes set $\mathcal{V}^* \subseteq \{\{e_u, e_v\} \mid (u, e_u), (v, e_v) \in \mathcal{E}', e_u \in \mathcal{V}_U, e_v \in \mathcal{V}_I\}$. Each attribute in $\mathcal{V}^*$ is an attribute entity from HIN $\mathcal{G}$, e.g., user's gender, item's genre. With a minimal set of $\mathcal{V}^*$, the counterfactual reasoning pursues to answer: what the fairness disparity would be, if $\mathcal{V}^*$ is applied to the recommendation model. $\mathcal{V}^*$ is recognized as a valid *counterfactual explanation for fairness*, if after applied $\mathcal{V}^*$, the fairness disparity of the intervened recommendation result $\Delta(H_{u,K}^{cf})$ reduced compared with original $\Delta(H_{u,K})$. In addition, $\mathcal{V}^*$ is *minimal* such that there is no smaller set $\mathcal{V}^{*'} \in \mathcal{G}$ satisfying $|\mathcal{V}^{*'}| < |\mathcal{V}^*|$ when $\mathcal{V}^{*'}$ is also valid.

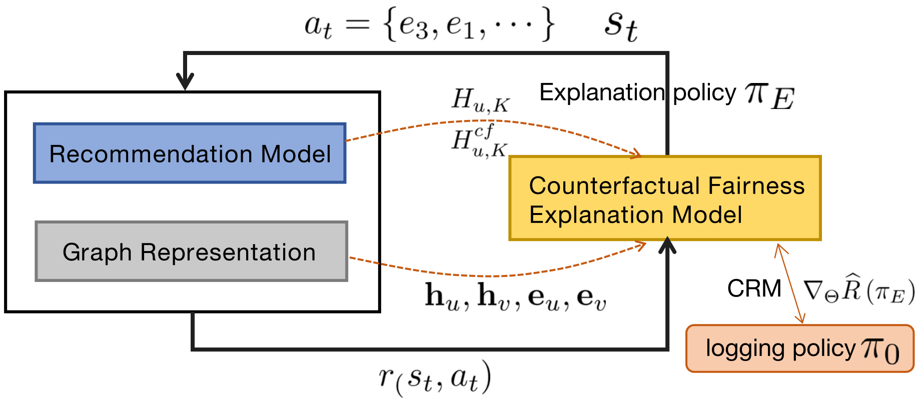

$a_t = \{e_3, e_1, \cdots\}$
s_t
$H_{u,K}$
$H^{cf}_{u,K}$
Explanation policy π_E
Recommendation Model
Graph Representation
Counterfactual Fairness Explanation Model
$\mathbf{h}_u, \mathbf{h}_v, \mathbf{e}_u, \mathbf{e}_v$
CRM
$\nabla_\Theta \widehat{R}(\pi_E)$
logging policy π_0
$r(s_t, a_t)$

Fig. 2. The proposed *CFairER* framework.

4 THE *CFAIRER* FRAMEWORK

We now introduce the framework of our *Counterfactual Explanation for Fairness (CFairER)*. As shown in Figure 2, *CFairER* devises three major components: 1) graph representation module embeds users, items, and attributes among HIN as embedding vectors; 2) recommendation model learns user and item latent factors to produce recommendation results and 3) our proposed counterfactual fairness explanation (CFE) model assisted by the graph representation module and the recommendation model to conduct counterfactual reasoning. This section discusses how the CFE model collaborates with the other two components, then introduces the graph representation module and the recommendation model. We will elaborate on our proposed CFE model in the next section.

4.1 Counterfactual Fairness Explanation Model

As shown in Figure 2, our CFE model is crafted within an off-policy learning environment, in which an explanation policy π_E is optimized to produce attribute-level counterfactual explanations for fairness. At each state s_t, π_E produces actions a_t absorbing user and item attributes as potential counterfactual explanations. These actions are committed to the recommendation model and graph representation module to produce the reward $r(s_t, a_t)$ for optimizing π_E. Specifically, the graph representation module provides dense vectors $\mathbf{h}_u$, $\mathbf{h}_v$, $\mathbf{e}_u$ and $\mathbf{e}_v$ as user, item, user attribute and item attribute embeddings, respectively. Those embeddings are used in the state representation learning (i.e., learn s_t) and attentive action pruning (i.e., select a_t) in our CFE model. Moreover, the attribute embeddings are fused with user or item latent factors learned by the recommendation model to explore the model fairness change. In particular, the fused embeddings of users and items are used to predict the intervened recommendation result $H^{cf}_{u,K}$. By comparing the fairness disparity (cf. Eq. (1)) difference between $H^{cf}_{u,K}$ and the original recommendation $H_{u,K}$, we determine the reward $r(s_t, a_t)$ to optimize π_E, accordingly. The reward $r(s_t, a_t)$ measures whether the current attribute (i.e., action) is a feasible fairness explanation responsible for the fairness change. Finally, π_E is optimized with a counterfactual risk minimization (CRM) objective $\nabla_\Theta \widehat{R}(\pi_E)$ to balance the distribution discrepancy from the logging policy π_0.

[1] Following [16], we consider the top 20% items with the most frequent interactions with users as G_0, while the remaining 80% belongs to G_1.

4.2 Graph Representation Module

Our graph representation module conducts heterogeneous graph representation learning to produce dense vectors of users, items, and attributes among the HIN. Compared with homogeneous graph learning such as GraphSage [21], our graph representation injects both node and edge heterogeneity to preserve the complex structure of the HIN. In particular, we include two weight matrices to specify varying weights of different node and edge types.

In the following, we present the graph learning for user embedding $\mathbf{h}_u$. The embeddings of $\mathbf{h}_v$, $\mathbf{e}_u$ and $\mathbf{e}_v$ can be obtained analogously by replacing nodes and node types while computations. Specifically, we first use Multi-OneHot [59] to initialize node embeddings at the 0-th layer, in which u's embedding is denoted by $\mathbf{h}_u^0$. Then, at each layer l, user embedding $\mathbf{h}_u^l$ is given by aggregating node u's neighbor information w.r.t. different node and edge types:

$$\mathbf{h}_u^l = \sigma\left(\text{concat}\left[\mathbf{W}_{\phi(u)}^l \, \mathrm{D}_p\left[\mathbf{h}_u^{l-1}\right], \frac{\mathbf{W}_{\psi(e)}^l}{\left|\mathcal{N}_{\psi(e)}(u)\right|} \sum_{u' \in \mathcal{N}_{\psi(e)}(u)} \mathrm{D}_p\left[\mathbf{h}_{u'}^{l-1}\right]\right] + b^l\right) \tag{2}$$

where $\sigma(\cdot)$ is LeakyReLU [55] activation function and concat$(\cdot)$ is the concatenation operator. $\mathrm{D}_p[\cdot]$ is a random dropout with probability p applied to its argument vector. $\mathbf{h}_u^{l-1}$ is u's embedding at layer $l-1$. $\mathcal{N}_{\psi(e)}(u) = \{u' \mid (u, e, u') \in \mathcal{G}\}$ is a set of nodes connected with user node u through edge type $\psi(e)$. The additionally dotted two weight matrices, i.e., node-type matrix $\mathbf{W}_{\phi(u)}^l$ and edge-type matrix $\mathbf{W}_{\psi(e)}^l$, are defined based on the importance of each type $\phi(u)$ and $\psi(e)$. b^l is an optional bias.

With Eq (2), we obtain u's embedding $\mathbf{h}_u^l$ at each layer $l \in \{1, \cdots, L\}$. We then adopt layer-aggregation [56] to concatenate u's embeddings at all layers into a single vector, i.e., $\mathbf{h}_u = \mathbf{h}_u^{(1)} + \cdots + \mathbf{h}_u^{(L)}$. Finally, we have user node u's embedding $\mathbf{h}_u$ through aggregation. The item embedding $\mathbf{h}_v$, user attribute embedding $\mathbf{e}_u$ and item attribute embedding $\mathbf{e}_v$ can be calculated analogously.

4.3 Recommendation Model

The recommendation model f_R is initialized using user-item interaction matrix $\boldsymbol{Y}$ to produce the Top-K recommendation result $H_{u,K}$ for all users. Here, we employ a linear and simple matrix factorization (MF) [35] as the recommendation model f_R. Particularly, MF initializes IDs of users and items as latent factors, and uses the inner product of user and item latent factors as the predictive function:

$$f_R(u, v) = \boldsymbol{U}_u^\top \boldsymbol{V}_v \tag{3}$$

where $\boldsymbol{U}_u$ and $\boldsymbol{V}_v$ denote d-dimensional latent factors for user u and item v, respectively. We use the cross-entropy [60] loss to define the objective function of the recommendation model:

$$\mathcal{L}_R = -\sum_{u, v, y_{uv} \in Y} y_{uv} \log f_R(u, v) + (1 - y_{uv}) \log\left(1 - f_R(u, v)\right) \tag{4}$$

After optimizing the loss function $\mathcal{L}_R$, we can use the trained user and item latent factors (i.e., $\boldsymbol{U}$, $\boldsymbol{V}$) to produce the original Top-K recommendation lists $H_{u,K}$ for all users $u \in \mathcal{U}$.

5 REINFORCEMENT LEARNING FOR COUNTERFACTUAL FAIRNESS EXPLANATION

We put forward our counterfactual fairness explanation (CFE) model (cf. Figure 3), assisted by graph representation module and recommendation model, to generate explanation policy π_E for item exposure fairness. The explanation policy π_E is optimized within off-policy learning to adaptively learn attributes responsible for fairness changes. In the

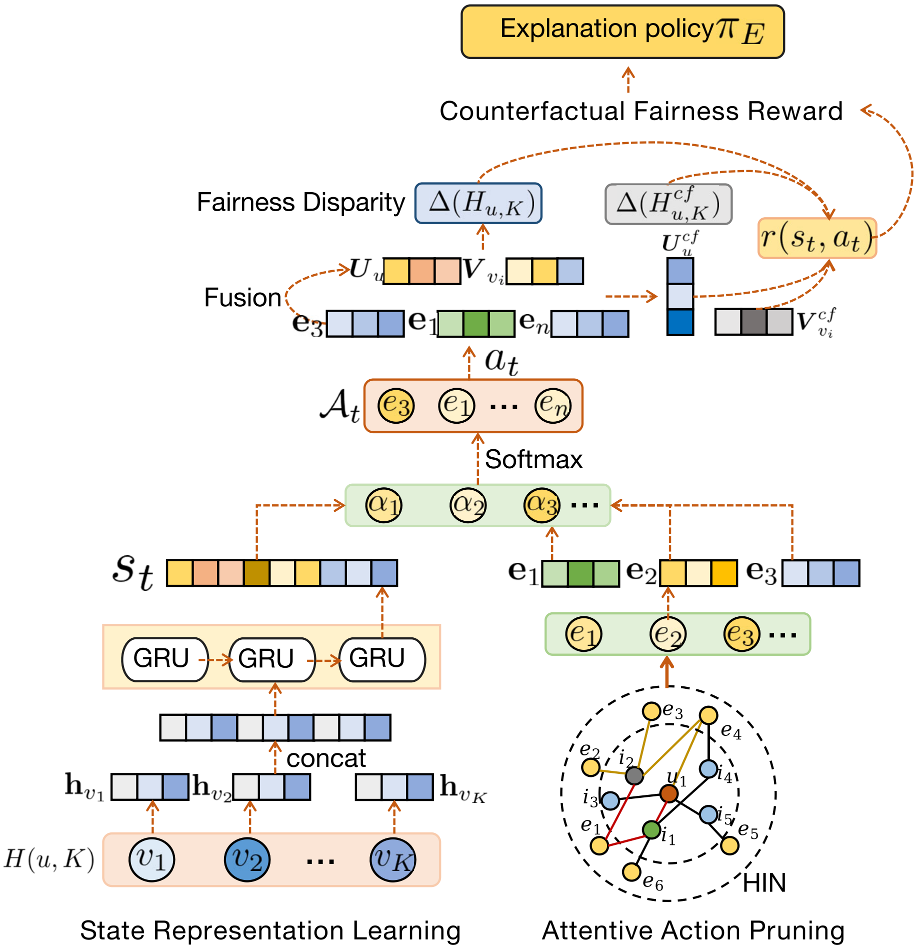


Fig. 3. Counterfactual Fairness Explanation (CFE) Model.

following, we first introduce off-policy learning for our CFE model. Then we detail each key element in the off-policy learning and give unbiased policy optimization.

5.1 Explaining as Off-policy Learning

We cast our CFE model in an off-policy learning environment, which is formulated as Markov Decision Process (MDP). The MDP is provided with a static logged dataset generated by a logging policy π_0 [2]. The logging policy π_0 collects trajectories by uniformly sampling actions from the user and item attribute space. We use the off-policy learning to optimize an explanation (i.e., target) policy π_E by approximating the counterfactual rewards of state-action pairs from all timestamps, wherein the logging policy π_0 is employed for exploration while the target policy π_E is utilized for decision-making. In the off-policy setting, the explanation policy π_E does not require following the original pace of the logging policy π_0. As a result, π_E is able to explore the counterfactual region, i.e., those actions that haven't been taken

[2] We adopt the uniform-based logging policy as π_0. It samples attributes as actions from the attribute space with the probability of $\pi_0(a_t \mid s_t) = \frac{1}{|\mathcal{V}_U + \mathcal{V}_I|}$.

by the previous agent using π_0. Formally, at each timestamp $t \in \{1, \cdots, T\}$ of MDP, the explanation policy $\pi_E(a_t|s_t)$ selects an action (i.e., a candidate attribute) $a_t \in \mathcal{A}_t$ conditioning on the user state $s_t \in \mathcal{S}$, and receives counterfactual reward $r(s_t, a_t) \in \mathcal{R}$ for this particular state-action pair. Then the current state transits to the next state s_{t+1} with transition probability of $\mathbb{P}(s_{t+1} \mid s_t, a_t) \in \mathcal{P}$. The whole MDP has the key elements:

- $\mathcal{S}$ is a finite set of states $\{s_t \mid t \in [1, \cdots, T]\}$. Each state s_t is transformed into dense vectors (i.e., embeddings) by our *state representation learning* (cf. Section 5.1.1).
- $\mathcal{A}_t$ is a finite set of actions (i.e., attributes) available at s_t. $\mathcal{A}_t$ is select from attributes $\mathcal{V}_t \in \mathcal{G}$ by our *attentive action pruning* (cf. Section 5.1.2) to reduce the search space.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the state transition, which absorbs transition probabilities of the current states to the next states. Given action a_t at state s_t, the transition to the next state s_{t+1} is determined as $\mathbb{P}(s_{t+1} \mid s_t, a_t) \in \mathcal{P} = 1$.
- $\mathcal{R} : \mathcal{S} \rightarrow \mathcal{R}$ is the counterfactual reward measures whether a deployed action (i.e., an attribute) is a valid counterfactual explanation for fairness. $\mathcal{R}$ is used to guide the explanation policy learning and is defined in Section 5.1.3.

We now introduce the implementation of each key component.

5.1.1 State Representation Learning. The state $\mathcal{S}$ describes target users and their recommendation lists from the recommendation model. Formally, at step t, the state s_t for a user u is defined as $s_t = (u, H(u, K))$, where $u \in \mathcal{U}$ is a target user and $H(u, K)$ is the recommendation produced by f_R. The initial state s_0 is (u, v) and v is the interacted item of u, i.e., $y_{uv} \in Y = 1$. Our state representation learning maps user state $s_t = (u, H(u, K))$ into dense vectors for latter explanation policy learning. Specifically, given s_t that absorbs current user u and its recommendation $H(u, K) = \{v_1, v_2, ..., v_K\}$. We first acquire the embedding $\mathbf{h}_{v_k}$ of each item $v_k \in H(u, K)$ from our graph representation module. The state s_t then receives the concatenated item embeddings (i.e., concat $\left[\mathbf{h}_{v_k} | \forall v_k \in H(u, K)\right]$) to update its representation. Considering states within $\mathcal{S}$ have sequential patterns [48], we resort to Recurrent Neural Networks (RNN) with a gated recurrent unit (GRU) [10] to capture the sequential state trajectory. We firstly initialize the state representation s_0 with an initial distribution $s_0 \sim \rho_0$ [3]. Then we learn state representation s_t through the recurrent cell:

$$
\begin{aligned}
\mathbf{u}_t &= \sigma_g \left(\mathbf{W}_1 \text{ concat} \left[\mathbf{h}_{v_k} | \forall v_k \in H(u, K)\right] + \mathbf{U}_1 s_{t-1} + b_1\right) \\
\mathbf{r}_t &= \sigma_g \left(\mathbf{W}_2 \text{ concat} \left[\mathbf{h}_{v_k} | \forall v_k \in H(u, K)\right] + \mathbf{U}_2 s_{t-1} + b_2\right) \\
\hat{s}_t &= \sigma_h \left(\mathbf{W}_3 \text{ concat} \left[\mathbf{h}_{v_k} | \forall v_k \in H(u, K)\right] + \mathbf{U}_3 \left(\mathbf{r}_t \cdot s_{t-1}\right) + b_3\right) \\
s_t &= (1 - \mathbf{u}_t) \cdot s_{t-1} + \mathbf{u}_t \odot \hat{s}_t
\end{aligned}
\tag{5}
$$

where $\mathbf{u}_t$ and $\mathbf{r}_t$ denote the update gate and reset gate vector generated by GRU and $\odot$ is the element-wise product operator. $\mathbf{W}_i$, $\mathbf{U}_i$ are weight matrices and b_i is the bias vector. Finally, s_t serves as the state representation at time step t.

5.1.2 Attentive Action Pruning. Our attentive action pruning is designed to reduce the action search space by specifying the varying importance of actions for each state. As a result, the sample efficiency can be largely increased by filtering out irrelevant actions to promote an efficient action search. In our method, actions are defined as candidate attributes selected from a given HIN that potentially impact the model fairness. In particular, given state $s_t = (u, H(u, K))$, we can distill a set of attributes $\mathcal{V}_t$ of the current user u and items $v \in H(u, K)$ from the HIN. Intuitively, we can directly use $\mathcal{V}_t$ as candidate actions for state s_t. However, the user and item attribute amount of the HIN would be huge, resulting in a large search space that terribly degrades the learning efficiency [47]. Thus, we propose an attentive action pruning

[3] In our experiment, we used a fixed initial state distribution, where $s_0 = 0 \in \mathbb{R}^d$

based on attention mechanism [42] to select important candidate actions for each state. Formally, given the embedding $\mathbf{e}_i$ for an attribute $i \in \mathcal{V}_t$ from Eq. (2), and the state representation s_t from Eq. (5), the attention score α_i of attribute i is:

$$\alpha_i = \text{ReLU}\left(\mathbf{W}_s s_t + \mathbf{W}_h \mathbf{e}_i + b\right) \tag{6}$$

where $\mathbf{W}_s$ and $\mathbf{W}_h$ are two weight matrices and b is the bias vector.

We then normalize attentive scores of all attributes in $\mathcal{V}_t$ and select attributes with n-top attention scores into $\mathcal{A}_t$:

$$\mathcal{A}_t = \left\{ i \mid i \in \text{Top-n}\left[\frac{\exp\left(\alpha_i\right)}{\sum_{i' \in \mathcal{V}_t} \exp\left(\alpha_{i'}\right)}\right] \text{ and } i \in \mathcal{V}_t \right\} \tag{7}$$

where n is the candidate size. To the end, our candidate set $\mathcal{A}_t$ is of high sample efficiency since it filters out irrelevant attributes while dynamically adapting to the user state shift.

5.1.3 Counterfactual Reward Definition. The counterfactual reward $r(s_t, a_t) \in \mathcal{R}$ measures whether a deployed action $a_t \in \mathcal{A}_t$ is a valid counterfactual explanation for fairness at the current state s_t. In particular, the reward is defined based on two criteria: 1) *Rationality* [43]: deploying action (i.e., attribute) a_t should cause the reduction of fairness disparity regarding the item exposure fairness. The fairness disparity change is measured by the fairness disparity difference between the recommendation result before (i.e., $\Delta(H_{u,K})$) and after (i.e., $\Delta(H^{cf}_{u,K})$) fusing the action a_t to the recommendation model f_R, i.e., $\Delta(H_{u,K}) - \Delta(H^{cf}_{u,K})$. 2) *Proximity* [12]: a counterfactual explanation is a minimal set of attributes that changes the fairness disparity.

For the *Rationality*, we fuse the embedding of a_t with user or item latent factors from the recommendation model to learn updated user and item latent vectors, so as to get the $\Delta(H^{cf}_{u,K})$. Specifically, for a state $s_t = (u, H(u, K))$, the embedding $\mathbf{e}_t$ of action a_t is fused to user latent factor $\boldsymbol{U}_u$ for user u and item latent factors $\boldsymbol{V}_{v_i}$ for all items $v_i \in H(u, K)$ by a element-wise product fusion. As a result, we can get the updated latent factors $\boldsymbol{U}^{cf}_u$ and $\boldsymbol{V}^{cf}_v$:

$$\begin{aligned} \boldsymbol{U}^{cf}_u &\leftarrow \boldsymbol{U}_u \odot \{\mathbf{e}_t \mid \forall t \in [1, \cdots, T]\}, \text{if } a_t \in \mathcal{V}_U \\ \boldsymbol{V}^{cf}_{v_i} &\leftarrow \boldsymbol{V}_{v_i} \odot \{\mathbf{e}_t \mid \forall t \in [1, \cdots, T]\}, \text{if } a_t \in \mathcal{V}_I \end{aligned} \tag{8}$$

where $\odot$ represents the element-wise product (a.k.a. Hadamard product). T is the total training iteration. At the initial state of $t = 0$, user and item latent factors $\boldsymbol{U}_u$ and $\boldsymbol{V}_v$ are learned form Eq (3). Through Eq. (8), the updated user and item latent vectors are then used to generate the intervened recommendation result $H^{cf}_{u,K}$.

For the *Proximity*, we compute whether a_t returns a minimal set of attributes that changes the recommendation model fairness. This is equal to regulating user and item latent factors before (i.e., $\boldsymbol{U}_u$, $\boldsymbol{V}_v$) and after (i.e., $\boldsymbol{U}^{cf}_u$, $\boldsymbol{V}^{cf}_v$) fusing a_t be as similar as possible.

Based on the two criteria, the counterfactual reward can be defined as the following form:

$$r(s_t, a_t) = \begin{cases} 1 + \text{dist}(\boldsymbol{U}_u, \boldsymbol{U}^{cf}_u) + \text{dist}(\boldsymbol{V}_v, \boldsymbol{V}^{cf}_v), \text{ if } \Delta(H_{u,K}) - \Delta(H^{cf}_{u,K}) \geq \epsilon \\ \text{dist}(\boldsymbol{U}_u, \boldsymbol{U}^{cf}_u) + \text{dist}(\boldsymbol{V}_v, \boldsymbol{V}^{cf}_v), \text{ otherwise} \end{cases} \tag{9}$$

where $\text{dist}(\cdot)$ is the distance metric defined as cosine similarity [31], i.e., $\text{dist}(a, b) = \frac{\langle a,b \rangle}{\|a\| \|b\|}$. $\Delta(\cdot)$ is the fairness disparity evaluation metric defined in Eq.(1). ϵ is the disparity change threshold that controls the model flexibility.

5.2 Unbiased Policy Optimization

Using state $s_t \in \mathcal{S}$ from Eq. (5), candidate action $a_t \in \mathcal{A}_t$ from Eq. (7), and counterfactual reward $r(s_t, a_t)$ in Eq. (9) for each timestamp t, the policy optimization seeks the explanation policy π_E that maximizes the expected cumulative reward $R(\pi_E)$ over total iteration T. Intuitively, we can directly use the policy gradient calculated on $R(\pi_E)$ to guide the optimization of π_E. However, our policy optimization is conducted in the off-policy learning setting, in which π_E holds different distribution from the logging policy π_0. Directly optimizing $R(\pi_E)$ would result in a biased policy optimization [47] due to the policy distribution discrepancy. To this end, we additionally apply *Counterfactual Risk Minimization* (CRM) [40] to correct the discrepancy between π_E and π_0. In particular, CRM employs an Inverse Propensity Scoring (IPS) [52] to explicitly estimate the distribution shift between π_E and π_0. After applying the CRM, we can alleviate the policy distribution bias by calculating the CRM-based expected cumulative reward $\widehat{R}(\pi_E)$:

$$\widehat{R}(\pi_E) = \mathbb{E}_{\pi_E}\left[\sum_{t=0}^{T} \gamma^t \frac{\pi_E\left(a_t \mid s_t\right)}{\pi_0\left(a_t \mid s_t\right)} r\left(s_t, a_t\right)\right] \tag{10}$$

where $\frac{\pi_E(a_t|s_t)}{\pi_0(a_t|s_t)}$ is called the *propensity score* for balancing the empirical risk estimated from the π_0.

Finally, the policy gradient of the explanation policy learning w.r.t. model parameter Θ is achieved by the REINFORCE [51]:

$$\nabla_\Theta \widehat{R}\left(\pi_E\right) = \frac{1}{T} \sum_{t=0}^{T} \gamma^t \frac{\pi_E\left(a_t \mid s_t\right)}{\pi_0\left(a_t \mid s_t\right)} r\left(s_t, a_t\right) \nabla_\Theta \log \pi_E(a_t \mid s_t) \tag{11}$$

where T is the total training iteration. By optimizing the Eq. (11), the learned explanation policy π_E generates minimal sets of attributes responsible for item exposure fairness changes, so as to find the true reasons leading to unfair recommendations.

6 EXPERIMENTS

We conduct extensive experiments to evaluate the proposed *CFairER* for explaining item exposure fairness in recommendations. We aim to particularly answer the following research questions:

- **RQ1.** Whether *CFairER* produces attribute-level explanations that are faithful to explaining recommendation model fairness compared with existing approaches?
- **RQ2.** Whether explanations provided by *CFairER* achieve better fairness-accuracy trade-off than other methods?
- **RQ3.** Do different components (i.e., attentive action pruning, counterfactual risk minimization-based optimization) help *CFairER* to achieve better sample efficiency and bias alleviation? How do hyper-parameters impact *CFairER*?

6.1 Experimental Setup

6.1.1 Datasets. We use logged user behavior data from three datasets `Yelp` [4], `Douban Movie` [5] and `Last-FM` [6] for evaluations. Each dataset is considered as an independent benchmark for different tasks, i.e., business, movie and music recommendation tasks. The `Yelp` dataset records user ratings on local businesses and business compliment, category and city profiles. The `Douban Movie` is a movie recommendation dataset that contains user group information and movie actor, director and type details. The `Last-FM` contains music listening records of users and artist tags. The details of both datasets are given in Table 1, which depicts statistics of user-item interactions, user-attribute and item-attribute relations. All datasets constitute complex user-item interactions and diverse attributes, thus providing rich contextual information for fairness explanation learning. Following previous works [25, 47, 49], we adopt a 10-core setting, i.e., retaining users and items with at least ten interactions for both datasets to ensure the dataset quality. Meanwhile, we

[4] https://www.yelp.com/dataset/
[5] https://movie.douban.com/
[6] https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding

binarize the explicit rating data by interpreting ratings of 4 or higher as positive feedback, otherwise negative. Then, we sort the interacted items for each user based on the timestamp and split the chronological interaction list into train/test/valid sets with a proportion of 60%/20%/20%.

Table 1. Statistics of the datasets.

Dataset		Yelp	Douban Movie	Last-FM
User-Item Interaction	#Users	16,239	13,367	1,892
	#Items	14,284	12,677	17,632
	#Interactions	198,397	1,068,278	92,834
	#Density	0.086%	0.63%	0.28%
User Item Attributes	#User Attributes	16,250	16,120	1,892
	#User-side Relations	235,465	574,132	25,434
	#Item Attributes	558	8,798	29,577
	#Item-side Relations	54,276	72,531	338,340

We also study the long-tail distribution of user-item interactions in the three datasets. We present the visualization results of the distribution of historical user-item interactions in the three datasets in Figure 4. Analyzing Figure 4, we find that user-item interactions of both datasets are presented with a skewed distribution: the head-tailed distribution in the blue plot area and the long-tailed distribution in the yellow plot area. Besides, a small fraction of popular items account for most of the user interactions in both datasets, The skewed distribution would result in serious item exposure unfairness issues in recommendations, such as the well-known filter-bubble problem [32] and Matthew effect [33].

6.1.2 Baselines. We adopt three heuristic approaches and two existing fairness-aware explainable recommendation methods as baselines. In particular,

- **RDExp**: We randomly select attributes from the attribute space for each user-item interaction and generate explanations based on the selected attributes. Note that the selected attributes can be both user and item attributes.
- **PopUser** and **PopItem**: We separately calculate the exposure number of attributes for each user-item interaction, then sort each attribute chronologically based on the exposure number. We devise a baseline **PopUser**, in which the top user attributes are selected as explanations. Analogously, we build **PopItem** that produces the top item attributes for the explanation.
- **FairKGAT**: uses FairKG4Rec [13] to mitigate the unfairness of explanations for a knowledge graph-enhanced recommender KGAT [44]. FairKG4Rec [13] is a generalized fairness-aware algorithm that controls the unfairness of explanation diversity in the recommendation model. KGAT [44] is a state-of-the-art knowledge graph-enhanced recommendation model that gives the best fairness performance in the original FairKG4Rec paper.
- **CEF** [16]: is the first work that explains fairness in recommendation. It generates feature-based explanations for item exposure unfairness by perturbing user and item features and searches for features that change the fairness disparity.

Note that to the best of our knowledge, **FairKGAT** [13] and **CEF** [16] are the only two existing methods designed for explainable fairness recommendation tasks.

Figure/long_tail_distribution_YP.png

(a) Interaction distribution in Yelp.

Figure/long_tail_distribution_DM.png

(b) Interaction distribution in Douban Movie.

6.1.3 *Explanation Faithfulness Evaluation.*

We adopt the widely used erasure-based evaluation criterion [15] in Explainable AI to evaluate the explanation faithfulness. The erasure-based evaluation identifies the contributions of explanations by measuring model performance changes after these explanations are removed. As a result, one can tell whether the model actually relied on these particular explanations to make a prediction, i.e., faithful to the model. In our experiments, we use the erasure-based evaluation to test (I) the recommendation performance change and (II) the recommendation fairness change after a set of attributes from the generated explanation is removed. By doing so, we can identify whether our explanations are faithful to recommendation performance and fairness disparity.

Following [15], we remove certain attributes from the generated explanations and evaluate the resulting recommendation performance. Therefore, in the starting evaluation point, we consider all attributes and add them to the user and item embeddings. We then remove certain attributes from the generated explanations to observe recommendation and fairness changes at later evaluation points. In particular, we first use historical user-item interactions to train a recommendation model through Eq. (4) to generate user and item embeddings. Then, we fuse all attribute embeddings from Eq. (2) with the trained user and item embeddings. The user and item embeddings after fusion are used to generate recommendation results at the starting evaluation point. Thereafter, we conduct counterfactual reasoning using our *CFairER* to generate attribute-level counterfactual explanations for model fairness. Those generated explanations are defined as the erasure set of attributes for each user/item. Finally, we exclude the erasure set from attribute space, and fuse the embeddings of attributes after erasure with the trained user and item embeddings to generate new recommendation results.

Given the recommendation results at each evaluation point, we use Normalized Discounted Cumulative Gain (NDCG)@K and Hit Ratio (HR)@K to measure the recommendation performance As this work focuses on item exposure fairness in recommendations, we use two wildly-adopted item-side evaluation metrics, i.e., Head-tailed Rate (HT)@K and Gini@K, for fairness evaluation. HT@K refers to the ratio of the head-tailed item number to the list length K. Later HT@K indicates that the model suffers from a more severe item exposure disparity by favoring items from the head-tailed (i.e., popular) group. Gini@K measures inequality within subgroups among the Top-K recommendation list. Larger Gini@K indicates the recommendation results are of higher inequality between the head-tailed and the long-tailed group.

6.1.4 *Implementation Details.*

To demonstrate our *CFairER*, we employ a simple matrix factorization (MF) as our recommendation model. We train the MF using train/test/validate sets split from user-item interactions in datasets with 60%/20%/20%. We optimize the MF using stochastic gradient descent (SGD) [5]. The same data splitting and gradient descent methods are applied in all baselines when required. Our graph representation module employs two graph convolutional layers with $\{64, 128\}$ output dimensions. FairKGAT baseline also keep 2 layers. The graph representation module outputs embeddings for all user and item attributes with the embedding size $d = 128$. The embedding size for FairKGAT and CEF is also fixed as $d = 128$. The number of latent factors (as in Eq. (3)) of MF is set equal to the embedding size of our graph representation module. To generate the starting evaluation point of erasure-based evaluation, we fuse attribute embeddings with the trained user and item latent factors based on element-wise product fusion. The fused user and item embeddings are then used to produce Top-K recommendation lists.

We train our counterfactual fairness explanation model with SGD based on the REINFORCE [39] policy gradient. For baseline model compatibility, as CEF [16] requires pre-defined user-feature attention matrix and item-feature quality matrix, we follow previous work [**?**] to regulate user/item attributes as user/item aspects and resort to analysis toolkit

"Sentires" [7] to build the two matrices. The hyper-parameters of our *CFairER* and all baselines are chosen by the grid search, including learning rate, L_2 norm regularization, discount factor γ, etc. The disparity change threshold ϵ in Eq. (9) of our *CFairER* is determined by performing a grid search on the validation set. This enables us to choose the optimal value for a variety of recommendation tasks, including but not limited to business (`Yelp` dataset), movie (`Douban Movie` dataset), and music (`Last-FM` dataset) recommendations. After all models have been trained, we freeze the model parameters and generate explanations accordingly. We report the erasure-based evaluation results by recursively erasing top E attributes from the generated explanations. The erasure length E is chosen from $E = [5, 10, 15, 20]$. The recommendation and fairness performance of our *CFairER* and baselines under different E is reported in Table 2.

6.2 Explanation Faithfulness (RQ1, RQ2)

Table 2. Recommendation and fairness performance after erasing top $E = [5, 10, 20]$ attributes from explanations. ↑ represents larger values are desired for better performance, while ↓ indicates smaller values are better. Bold and underlined numbers are the best results and the second-best results, respectively.

Method	NDCG@40 ↑			Hit Ratio (HR)@40 ↑			Head-tailed Rate (HT)@40 ↓			Gini@40 ↓		
	$E = 5$	$E = 10$	$E = 20$	$E = 5$	$E = 10$	$E = 20$	$E = 5$	$E = 10$	$E = 20$	$E = 5$	$E = 10$	$E = 20$
`Yelp`												
RDExp	0.0139	0.0125	0.0118	0.1153	0.1036	0.1029	0.1994	0.1976	0.1872	0.3870	0.3894	0.3701
PopUser	0.0141	0.0136	0.0128	0.1183	0.1072	0.1067	0.1776	0.1767	0.1718	0.3671	0.3642	0.3495
PopItem	0.0147	0.0139	0.0131	0.1182	0.1093	0.1084	0.1793	0.1846	0.1848	0.3384	0.3370	0.3359
FairKGAT	0.0153	0.0141	0.0138	0.1384	0.1290	0.1277	0.1802	0.1838	0.1823	0.3671	0.3508	0.3542
CEF	0.0254	0.0247	0.0231	0.1572	0.1608	0.1501	0.1496	0.1455	0.1420	0.3207	0.3159	0.3088
CFairER	**0.0316**	**0.0293**	**0.0291**	**0.1987**	**0.1872**	**0.1868**	**0.1345**	**0.1322**	**0.1301**	**0.2366**	**0.2068**	**0.1974**
`Douban Movie`												
RDExp	0.0390	0.0346	0.0351	0.1278	0.1170	0.1172	0.1932	0.1701	0.1693	0.3964	0.3862	0.3741
PopUser	0.0451	0.0352	0.0348	0.1482	0.1183	0.1174	0.1790	0.1674	0.1658	0.3684	0.3591	0.3562
PopItem	0.0458	0.0387	0.0379	0.1523	0.1219	0.1208	0.1831	0.1458	0.1383	0.3664	0.3768	0.3692
FairKGAT	0.0534	0.0477	0.0421	0.1602	0.1377	0.1308	0.1654	0.1573	0.1436	0.3590	0.3472	0.3483
CEF	0.0831	0.0795	0.0809	0.1949	0.1973	0.1901	0.1043	**0.0998**	**0.0945**	0.3079	0.2908	0.3001
CFairER	**0.1290**	**0.0921**	**0.0901**	**0.2706**	**0.2441**	**0.2238**	**0.0841**	0.1183	0.1101	**0.2878**	**0.2648**	**0.2593**
`Last-FM`												
RDExp	0.0857	0.0568	0.0592	0.8436	0.7915	0.7831	0.7884	0.7732	0.7691	0.3707	0.3531	0.3540
PopUser	0.0786	0.0432	0.0431	0.7787	0.7697	0.7604	0.7979	0.8064	0.7942	0.3862	0.3729	0.3761
PopItem	0.0792	0.0479	0.0435	0.7803	0.7961	0.7914	0.7689	0.7638	0.7573	0.3673	0.3602	0.3618
FairKGAT	0.0832	0.0594	0.0621	0.8063	0.7938	0.8165	0.7451	0.7342	0.7408	0.3580	0.3458	0.3433
CEF	0.0962	0.1037	0.1001	0.8592	0.8408	0.8509	0.6873	0.6092	**0.5601**	0.3308	0.3298	0.3375
CFairER	**0.1333**	**0.1193**	**0.1187**	**0.9176**	**0.8867**	**0.8921**	**0.6142**	**0.5865**	0.5737	**0.2408**	**0.2371**	**0.2385**

We plot fairness and recommendation performance changes of our *CFairER* and baselines while erasing attributes from explanations in Figure 5. Each data point in Figure 5 is generated by cumulatively erasing a batch of attributes. Those erased attributes are selected from the top 10 (i.e., $E = 10$) attribute sets of the explanation lists provided by each method.[8] As PopUser and PopItem baselines enjoy very similar data trends, we choose not to present them simultaneously in Figure 5. Table 2 presents recommendation and fairness performance after erasing $E = [5, 10, 20]$

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

[8] For example, given n explanation lists, the number of erasure attributes is $n \times 10$. We cumulatively erase m attributes in one batch within in total $(n \times 10)/m$ iterations.

attributes in explanations. Larger NDCG@K and Hit Ratio @K values indicate better recommendation performance while smaller Head-tailed Rate@K and Gini@K values represent better fairness. Analyzing Figure 5 and Table 2, we have the following findings.

Amongst all methods, our *CFairER* achieves the best recommendation and fairness performance after erasing attributes from our explanations on all datasets. For instance, *CFairER* beats the strongest baseline CEF by 25.9%, 24.4%, 8.3% and 36.0% for NDCG@40, Hit Ratio@40, Head-tailed Rate@40 and Gini@40 with erasure length $E = 20$ on `Yelp`. This indicates that explanations generated by *CFairER* are faithful to explaining unfair factors while not harming recommendation accuracy. Unlike CEF and FairKGAT, which generate explanations based on perturbing input features and adding fair-related constraints, *CFairER* generates counterfactual explanations by inferring minimal attributes contributing to fairness changes. As a counterfactual explanation is minimal, it only discovers attributes that well-explain the model fairness while filtering out tedious ones that affect the recommendation accuracy. Another interesting finding is that PopUser and PopItem perform even worse than RDExp (i.e., randomly selecting attributes) on `Last-FM`. This is because recommending items with popular attributes would deprive the exposure of less-noticeable items, causing serious model unfairness and degraded recommendation performance.

In general, the fairness of all models consistently improves while erasing attributes from explanations, shown by the decreasing trend of Head-tailed Rate@K values in Figure 5. This is because erasing attributes will alleviate the discrimination against users and items from disadvantaged groups (e.g., gender group, brand group), making more under-represented items to be recommended. Unfortunately, we can also observe the downgraded recommendation performance of all models in both Figure 5 and Table 2. For example, in Figure 5, the NDCG@5 of CEF drops from approximately 1.17 to 0.60 on `Last-FM` at erasure iteration 0 and 50. This is due to the well-known fairness-accuracy trade-off issue, in which the fairness constraint could be achieved with a sacrifice of recommendation performance. Facing this issue, both baselines suffer from huge declines in recommendation performance, as in Table 2. On the contrary, our *CFairER* still enjoys favorable recommendation performance and outperforms all baselines. Besides, the decline rates of our *CFairER* are much slower than baselines on both datasets in Figure 5. We hence conclude that the attribute-level explanations provided by our *CFairER* can achieve a much better fairness-accuracy trade-off than other methods. This is because our *CFairER* uses counterfactual reasoning to generate minimal but vital attributes as explanations for model fairness. Those attributes produced by *CFairER* are true reasons for unfairness but not the ones that affect the recommendation accuracy.

6.3 Ablation and Parameter Analysis (RQ3)

We first conduct an in-depth ablation study on the ability of our *CFairER* to achieve sample efficiency and bias alleviation. Our *CFairER* includes two contributing components, namely, attentive action pruning (cf. Section 5.1) and counterfactual risk minimization-based optimization (cf. Section 5.2). We evaluate our *CFairER* with different variant combinations and show our main findings below.

6.3.1 Sample Efficiency of Attentive Action Pruning. Our attentive action pruning reduces the action search space by specifying varying importance of attributes for each state. As a result, the sample efficiency can be increased by filtering out irrelevant attributes to promote an efficient action search. To demonstrate our attentive action pruning, we test *CFairER* without (¬) the attentive action pruning (i.e., *CFairER ¬ Attentive Action Pruning*), in which the candidate actions set absorbs all attributes connected with the current user and items. Through Table 3, we observed that removing the attentive action pruning downgrades *CFairER* performance, which validates the superiority of our attentive action

Table 3. Ablation Study on *CFairER*. Erasure length E is fixed as $E = 20$. ¬ represents the corresponding module is removed. $A \longrightarrow B$ represents A is replaced by B. ± indicates the increase or decrease percentage of the variant compared with *CFairER*.

Variants	NDCG@20↑	HR@20 ↑	HT@20↓	Gini@20↓
Yelp				
CFairER	0.0238	0.1871	0.1684	0.1990
CFairER ¬ Attentive Action Pruning	0.0164(−31.1%)	0.1682(−10.1%)	0.1903(+13.0%)	0.2159(+8.5%)
CRM loss ⟶ Cross-entropy [60] loss	0.0197(−17.2%)	0.1704(−8.9%)	0.1841(+9.3%)	0.2101(+5.6%)
Douban Movie				
CFairER	0.0583	0.2043	0.1149	0.2871
CFairER ¬ Attentive Action Pruning	0.0374(−35.9%)	0.1537(−24.8%)	0.1592(+38.6%)	0.3574(+24.5%)
CRM loss ⟶ Cross-entropy [60] loss	0.0473(−18.9%)	0.1582(−22.6%)	0.1297(+12.9%)	0.3042(+6.0%)
Last-FM				
CFairER	0.1142	0.7801	0.6914	0.2670
CFairER ¬ Attentive Action Pruning	0.0987(−13.6%)	0.7451(−4.5%)	0.7833(+13.3%)	0.2942(+10.2%)
CRM loss ⟶ Cross-entropy [60] loss	0.0996(−12.8%)	0.7483(−4.1%)	0.7701(+11.4%)	0.2831(+6.0%)

pruning in improving fair recommendations. This is because attentive action pruning filters out irrelevant items based on their contributions to the current state, resulting in enhanced sample efficiency. Moreover, the performance of *CFairER* after removing the attentive action pruning downgrades severely on `Douban Movie`. This is because `Douban Movie` has the largest number of attributes compared with the other two datasets (cf. Table 1), which challenges our *CFairER* to find suitable attributes as fairness explanations. These findings suggest the superiority of applying attentive action pruning in fairness explanation learning, especially when the attribute size is large.

6.3.2 Bias Alleviation with Counterfactual Risk Minimization. Our *CFairER* is optimized with a counterfactual risk minimization (CRM) loss to achieve unbiased policy optimization. The CRM loss (cf. Eq. (10)) corrects the discrepancy between the explanation policy and logging policy, thus alleviating the policy distribution bias in the off-policy learning setting. To demonstrate the CRM loss, we apply our *CFairER* with cross-entropy (CE) [60] loss (i.e., *CRM loss ⟶ Cross-entropy loss*) to show how it performs compared with *CFairER* on the CRM loss. We observe our *CFairER* with CRM loss consistently outperforms the counterpart with CE loss on both fairness and recommendation performance. The sub-optimal performance of our *CFairER* with CE loss indicates that the bias issue in the off-policy learning can lead to downgraded performance for the learning agent. On the contrary, our *CFairER* takes advantage of CRM to learn a high-quality explanation policy. We hence conclude that performing unbiased optimization with CRM is critical to achieving favorable fairness explanation learning.

6.3.3 Parameter Analysis. We also conduct a parameter analysis on how erasure length E (cf. Section 6.1.3) and candidate size n (as in Eq. (7)) impact *CFairER*. Figure 6 (a) and Figure 6 (b) report *CFairER* performance w.r.t. $E = [5, 10, 15, 20]$. Apparently, the performance of *CFairER* demonstrates decreasing trends from $E = 5$, then becomes stable after $E = 10$. The decreased performance is due to the increasing erasure of attributes found by our generated explanations. This indicates that our *CFairER* can find valid attribute-level explanations that impact fair recommendations. The performance of *CFairER* degrades slightly after the bottom, then becomes stable. This is reasonable since the attributes number provided in datasets are limited, while increasing the erasure length would allow more overlapping attributes with previous erasures to be found.

By varying candidate size n from $n = [10, 20, 30, 40, 50, 60]$ in Figure 6 (c) (d), we observe that *CFairER* performance first improves drastically as candidate size increases on both datasets. The performance of our *CFairER* reaches peaks at $n = 40$ and $n = 30$ on `Yelp` and `Last-FM`, respectively. After the peaks, we can witness a downgraded model performance by increasing the candidate size further. We consider the poorer performance of *CFairER* before reaching peaks is due to the limited candidate pool, i.e., insufficient attributes limit the exploration ability of *CFairER* to find appropriate candidates as fairness explanations. Meanwhile, a too-large candidate pool (e.g., $n = 60$) would offer more chances for the agent to select inadequate attributes as explanations. Based on the two findings, we believe it is necessary for our *CFairER* to carry the attentive action search, such as to select high-quality attributes as candidates based on their contributions to the current state.

6.3.4 Time Complexity and Computation Costs. For time complexity, our recommendation model (cf. Section 4.3) performs matrix factorization with a complexity of $O(|\mathcal{O}|)$. For the graph representation module (cf. Section 4.2), establishing node representations has complexity $O(\sum_{l=1}^{L}(|\mathcal{G}| + |\mathcal{O}^+|)d_l d_{l-1})$. For the off-policy learning process (cf. Section 5.1), the complexity is mainly determined by the attention score calculation, which has a time complexity of $O(2T|\mathcal{O}^+||\tilde{\mathcal{N}}_e|d^2)$. The total time complexity is $O(|\mathcal{O}| + \sum_{l=1}^{L}(|\mathcal{G}| + |\mathcal{O}^+|)d_l d_{l-1} + 2T|\mathcal{O}^+|n_2 d^2)$. We evaluated the running time of FairKGAT and CEF baselines on the large-scale `Yelp` dataset. The corresponding results are 232s and 379s per epoch, respectively. *CFairER* has a comparable cost of 284s per epoch to these baselines. Considering that our *CFairER* achieves superior explainability improvements compared to the baselines, we believe that the increased cost of, at most, 52s per epoch is a reasonable trade-off.

7 CONCLUSION

We propose *CFairER*, a reinforcement learning-based fairness explanation learning framework over a HIN. Our *CFairER* generates counterfactual explanations as minimal sets of real-world attributes to explain item exposure fairness. We design a counterfactual fairness explanation model to discover high-quality counterfactual explanations, driven by an attentive action pruning to reduce the search space and a counterfactual reward to enable counterfactual reasoning. Extensive evaluations on three benchmark datasets demonstrate *CFairER*'s ability to find faithful explanations for fairness and balance the fairness-accuracy trade-off.

ACKNOWLEDGMENTS

This work is supported by the Australian Research Council (ARC) under Grant No. DP220103717, LE220100078, LP170100891 and DP200101374.

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

[3] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. 2020. Privacy-aware recommendation with private-attribute protection using adversarial learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 34–42.

[4] 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 International Conference on Knowledge Discovery & Data Mining*. 2212–2220.

[5] Léon Bottou. 2012. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*. Springer, 421–436.

[6] Ruth MJ Byrne. 2007. *The rational imagination: How people create alternatives to reality.* MIT press.

[7] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*. 1583–1592.

[8] Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. 2018. Investigating the impact of gender on rank in resume search engines. In *Proceedings of the 2018 chi conference on human factors in computing systems*. 1–14.

[9] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[10] Rahul Dey and Fathi M Salem. 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 1597–1600.

[11] Fernando Diaz, Bhaskar Mitra, Michael D Ekstrand, Asia J Biega, and Ben Carterette. 2020. Evaluating stochastic rankings with expected exposure. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 275–284.

[12] Carlos Fernandez, Foster J. Provost, and Xintian Han. 2020. Explaining Data-Driven Decisions made by AI Systems: The Counterfactual Approach. *CoRR* abs/2001.07417 (2020). arXiv:2001.07417 https://arxiv.org/abs/2001.07417

[13] 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 International ACM SIGIR Conference on Research and Development in Information Retrieval*. 69–78.

[14] 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.

[15] Yingqiang Ge, Shuchang Liu, Zelong Li, Shuyuan Xu, Shijie Geng, Yunqi Li, Juntao Tan, Fei Sun, and Yongfeng Zhang. 2021. Counterfactual Evaluation for Explainable AI. *arXiv preprint arXiv:2109.01962* (2021).

[16] Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022. Explainable Fairness in Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 681–691. https://doi.org/10.1145/3477495.3531973

[17] Yingqiang Ge, Xiaoting Zhao, Lucia Yu, Saurabh Paul, Diane Hu, Chu-Cheng Hsieh, and Yongfeng Zhang. 2022. Toward Pareto Efficient Fairness-Utility Trade-off in Recommendation through Reinforcement Learning. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 316–324.

[18] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 196–204.

[19] Mahesh Goyani and Neha Chaurasiya. 2020. A review of movie recommendation system: Limitations, Survey and Challenges. *ELCVIA: electronic letters on computer vision and image analysis* 19, 3 (2020), 0018–37.

[20] Ziyu Guan, Can Wang, Jiajun Bu, Chun Chen, Kun Yang, Deng Cai, and Xiaofei He. 2010. Document recommendation in social tagging services. In *Proceedings of the 19th international conference on World wide web*. 391–400.

[21] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[22] Li He, Hongxu Chen, Dingxian Wang, Shoaib Jameel, Philip Yu, and Guandong Xu. 2021. Click-Through Rate Prediction with Multi-Modal Hypergraphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 690–699.

[23] 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*. 173–182.

[24] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1531–1540.

[25] Qian Li, Xiangmeng Wang, Zhichao Wang, and Guandong Xu. 2022. Be Causal: De-Biasing Social Network Confounding in Recommendation. *ACM Trans. Knowl. Discov. Data* (apr 2022). https://doi.org/10.1145/3533725

[26] Qian Li, Zhichao Wang, Shaowu Liu, Gang Li, and Guandong Xu. 2022. Deep treatment-adaptive network for causal inference. *The VLDB Journal* (2022), 1–16.

[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, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2022. Fairness in Recommendation: A Survey. *arXiv preprint arXiv:2205.13619* (2022).

[29] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards personalized fairness based on causal notion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1054–1063.

[30] Weiwen Liu, Feng Liu, Ruiming Tang, Ben Liao, Guangyong Chen, and Pheng Ann Heng. 2020. Balancing between accuracy and fairness for interactive recommendation with reinforcement learning. In *Pacific-asia conference on knowledge discovery and data mining*. Springer, 155–167.

[31] Hieu V Nguyen and Li Bai. 2010. Cosine similarity metric learning for face verification. In *Asian conference on computer vision*. Springer, 709–720.

[32] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. 677–686.

[33] Matjaž Perc. 2014. The Matthew effect in empirical data. *Journal of The Royal Society Interface* 11, 98 (2014), 20140378.

[34] SRS Reddy, Sravani Nalluri, Subramanyam Kunisetti, S Ashok, and B Venkatesh. 2019. Content-based movie recommendation system using genre correlation. In *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 2*. Springer, 391–397.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[36] Yehezkel S Resheff, Yanai Elazar, Moni Shahar, and Oren Sar Shalom. 2018. Privacy and fairness in recommender systems via adversarial training of user representations. *arXiv preprint arXiv:1807.03521* (2018).

[37] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2018), 357–370.

[38] Chuan Shi, Zhiqiang Zhang, Yugang Ji, Weipeng Wang, Philip S Yu, and Zhiping Shi. 2019. SemRec: a personalized semantic recommendation method based on weighted heterogeneous information networks. *World Wide Web* 22 (2019), 153–184.

[39] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems* 12 (1999).

[40] Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*. PMLR, 814–823.

[41] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual Explanations for Neural Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1627–1631.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[43] Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596* (2020).

[44] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

[45] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.

[46] Xiangmeng Wang, Qian Li, Dianer Yu, Peng Cui, Zhichao Wang, and Guandong Xu. 2022. Causal Disentanglement for Semantics-Aware Intent Learning in Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[47] Xiangmeng Wang, Qian Li, Dianer Yu, Zhichao Wang, Hongxu Chen, and Guandong Xu. 2022. MGPolicy: Meta Graph Enhanced Off-Policy Learning for Recommendations. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1369–1378. https://doi.org/10.1145/3477495.3532021

[48] Xiangmeng Wang, Qian Li, Dianer Yu, and Guandong Xu. 2022. Off-Policy Learning over Heterogeneous Information for Recommendation *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 2348–2359. https://doi.org/10.1145/3485447.3512072

[49] Xiangmeng Wang, Qian Li, Wu Zhang, Guandong Xu, Shaowu Liu, and Wenhao Zhu. 2020. Joint relational dependency learning for sequential recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 168–180.

[50] Yifan Wang, Weizhi Ma, Min Zhang*, Yiqun Liu, and Shaoping Ma. 2022. A Survey on the Fairness of Recommender Systems. *ACM Journal of the ACM (JACM)* (2022).

[51] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[52] Elizabeth J Williamson and Andrew Forbes. 2014. Introduction to propensity scores. *Respirology* 19, 5 (2014), 625–635.

[53] James Woodward. 2004. *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, New York. https://doi.org/10.1093/0195155270.001.0001

[54] Kun Xiong, Wenwen Ye, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, Binbin Hu, Zhiqiang Zhang, and Jun Zhou. 2021. Counterfactual Review-based Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2231–2240.

[55] Jin Xu, Zishan Li, Bowen Du, Miaomiao Zhang, and Jing Liu. 2020. Reluplex made more practical: Leaky ReLU. In *2020 IEEE Symposium on Computers and communications (ISCC)*. IEEE, 1–7.

[56] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*. PMLR, 5453–5462.

[57] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. *Advances in neural information processing systems* 30 (2017).

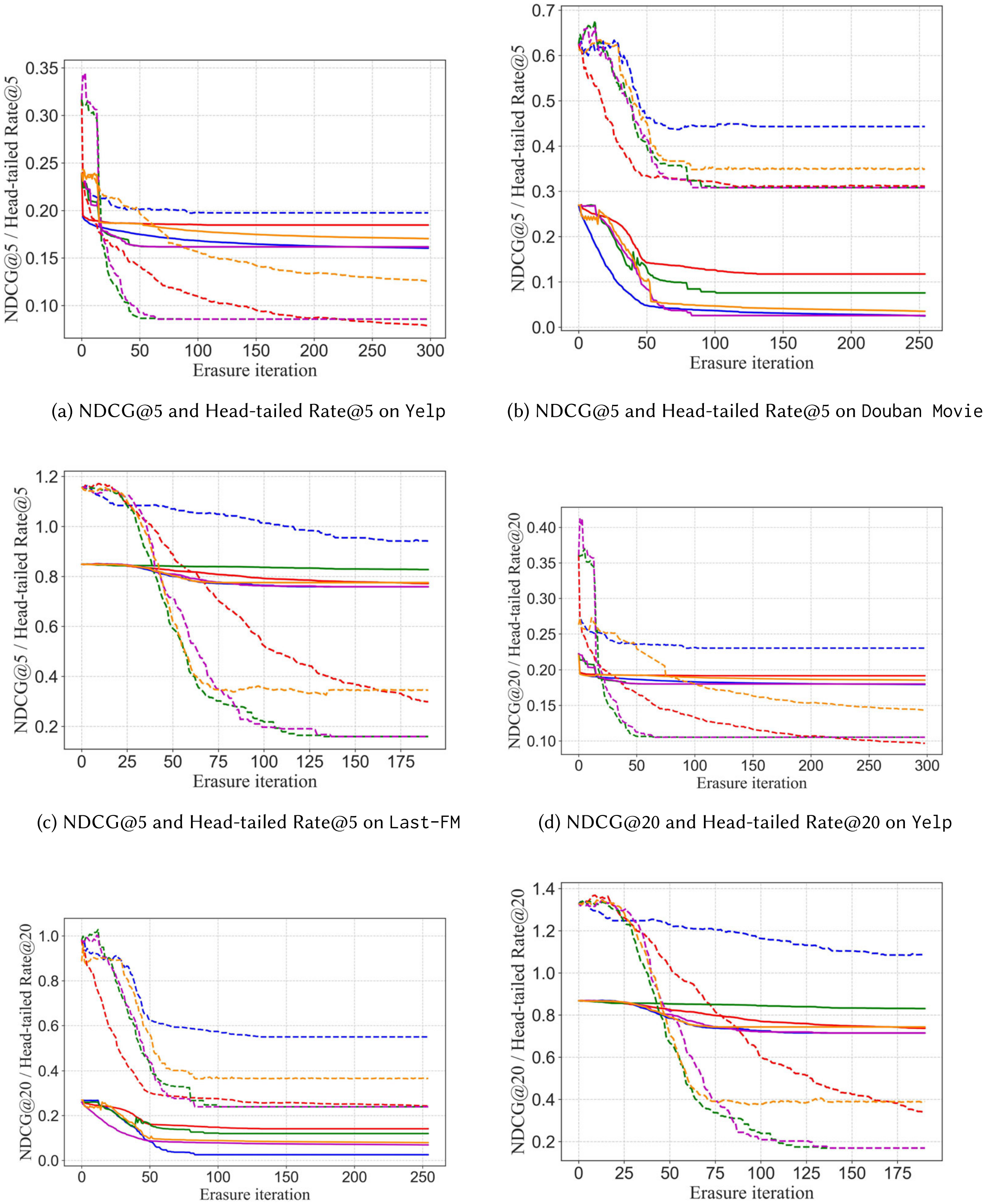
[58] Dianer Yu, Qian Li, Xiangmeng Wang, Zhichao Wang, Yanan Cao, and Guandong Xu. 2022. Semantics-Guided Disentangled Learning for Recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 249–261.

[59] Suiyun Zhang, Zhizhong Han, Yu-Kun Lai, Matthias Zwicker, and Hui Zhang. 2019. Stylistic scene enhancement GAN: mixed stylistic enhancement generation for 3D indoor scenes. *The Visual Computer* 35, 6 (2019), 1157–1169.

[60] Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems* 31 (2018).

[61] Lili Zhao, Zhongqi Lu, Sinno Jialin Pan, Qiang Yang, and Wei Xu. 2016. Matrix factorization+ for movie recommendation.. In *IJCAI*. 3945–3951.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009



(a) NDCG@5 and Head-tailed Rate@5 on `Yelp`

(b) NDCG@5 and Head-tailed Rate@5 on `Douban Movie`

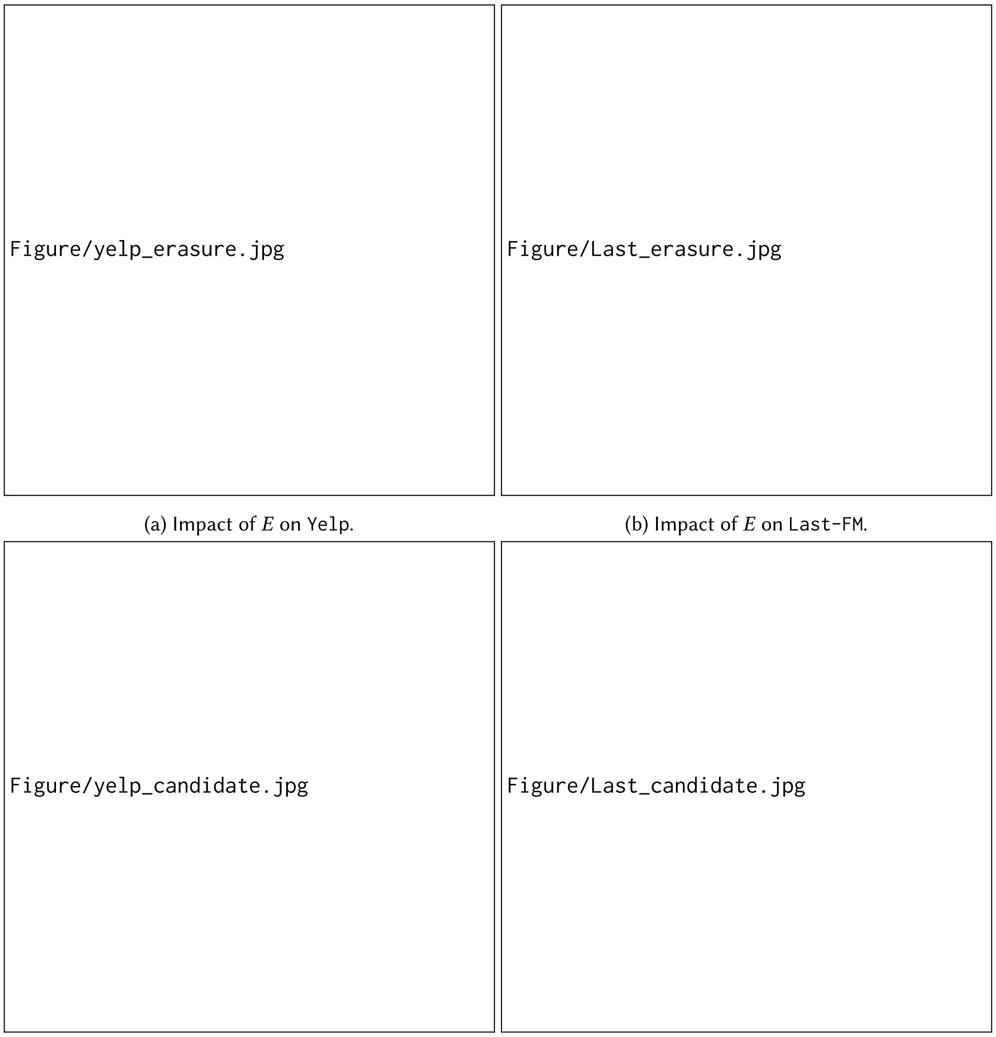
(c) NDCG@5 and Head-tailed Rate@5 on `Last-FM`

(d) NDCG@20 and Head-tailed Rate@20 on `Yelp`

(e) NDCG@20 and Head-tailed Rate@20 on `Douban Movie`

(f) NDCG@20 and Head-tailed Rate@20 on `Last-FM`

Fig. 5. Erasure-based evaluation on Top-5 and Top-20 recommendations. NDCG@K values reveal the recommendation performance of models, while Head-tailed Rate@K values reflect model fairness. NDCG@K values are multiplied with 10 for better presentation. Each data point is generated while cumulatively erasing the top 10 (i.e., $E = 10$) attributes in explanations.



(a) Impact of E on `Yelp`.

(b) Impact of E on `Last-FM`.

(c) Impact of n on `Yelp`.

(d) Impact of n on `Last-FM`.

Fig. 6. Impacts of parameters E and n on *CFairER*.