Deconfounded Recommendation for Alleviating Bias Amplification

Wenjie Wang[1], Fuli Feng[12*], Xiangnan He[3], Xiang Wang[12], and Tat-Seng Chua[1]
[1]National University of Singapore, [2]Sea-NExT Joint Lab, [3]University of Science and Technology of China
{wenjiewang96,fulifeng93,xiangnanhe}@gmail.com,xiangwang@u.nus.edu,dcscts@nus.edu.sg

ABSTRACT

Recommender systems usually amplify the biases in the data. The model learned from historical interactions with imbalanced item distribution will amplify the imbalance by over-recommending items from the majority groups. Addressing this issue is essential for a healthy ecosystem of recommendation in the long run. Existing work applies bias control to the ranking targets (*e.g.,* calibration, fairness, and diversity), but ignores the true reason for bias amplification and trades off the recommendation accuracy.

In this work, we scrutinize the cause-effect factors for bias amplification, identifying the main reason lies in the confounding effect of imbalanced item distribution on user representation and prediction score. The existence of such confounder pushes us to go beyond merely modeling the conditional probability and embrace the causal modeling for recommendation. Towards this end, we propose a *Deconfounded Recommender System* (DecRS), which models the causal effect of user representation on the prediction score. The key to eliminating the impact of the confounder lies in *backdoor adjustment*, which is however difficult to do due to the infinite sample space of the confounder. For this challenge, we contribute an approximation operator for backdoor adjustment which can be easily plugged into most recommender models. Lastly, we devise an inference strategy to dynamically regulate backdoor adjustment according to user status. We instantiate DecRS on two representative models FM [32] and NFM [16], and conduct extensive experiments over two benchmarks to validate the superiority of our proposed DecRS.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Collaborative filtering**.

KEYWORDS

Deconfounded Recommendation, User Interest Imbalance, Bias Amplification

∗ Corresponding author: Fuli Feng (fulifeng93@gmail.com). This research is supported by the Sea-NExT Joint Lab, the National Natural Science Foundation of China (61972372), and National Key Research and Development Program of China (2020AAA0106000).

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467249

ACM Reference Format:
Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467249

1 INTRODUCTION

Recommender System (RS) has been widely used to achieve personalized recommendation in most online services, such as social networks and advertising [25, 42]. Its default choice is to learn user interest from historical interactions (*e.g.,* clicks and purchases), which typically exhibit data bias, *i.e.,* the distribution over item groups (*e.g.,* the genre of movies) is imbalanced. Consequently, recommender models face the *bias amplification* issue [35]: over-recommending the majority group and amplifying the imbalance. Figure 1(a) illustrates this issue with an example in movie recommendation, where 70% of the movies watched by a user are action movies, but action movies take 90% of the recommendation slots. Undoubtedly, over-emphasizing the items from the majority groups will limit a user's view and decrease the effectiveness of recommendations. Worse still, due to feedback loop [7], such bias amplification will intensify with time, causing more issues like filter bubbles [23] and echo chambers [14].

Existing work alleviates bias amplification by introducing bias control into the ranking objective of recommender models, which is mainly from three perspectives: 1) fairness [22, 34], which pursues equal exposure opportunities for items of different groups; 2) diversity [6], which intentionally increases the covered groups in a recommendation list, and 3) calibration [35], which encourages the distribution of recommended item groups to follow that of interacted items of the user. However, these methods alleviate bias amplification at the cost of sacrificing recommendation accuracy [34, 35]. More importantly, the fundamental question is not answered: what is the root reason for bias amplification?

After inspecting the cause-effect factors in recommender modeling, we attribute bias amplification to a *confounder* [28]. The historical distribution of a user over item groups (*e.g.,* [0.7, 0.3] in Figure 1(a)) is a confounder between the user's representation and the prediction score. In the conventional RS, the user/item representations are then fed into an interaction module (*e.g.,* factorization machines (FM) [32]) to calculate the prediction score [17]. In other words, recommender models estimate the *conditional probability* of clicks given user/item representations. From a causal view, user and item representations are the causes of the prediction score, and the user historical distribution over item groups affects both the user representation and the prediction score. Inevitably, such hidden confounder makes the modeling of conditional probability suffer from a spurious correlation between

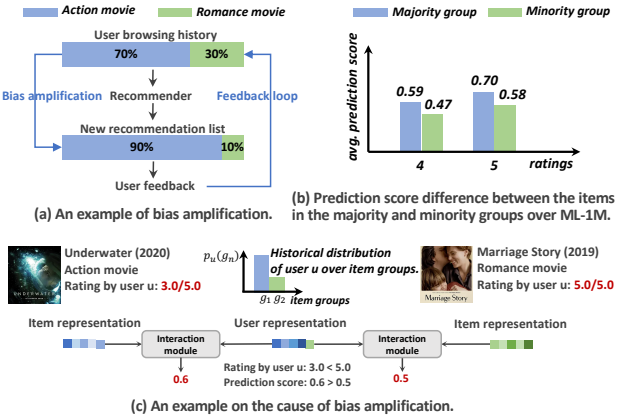


Figure 1: Illustration of bias amplification.

the user and the prediction score. That is, given two item groups, the one that the user interacted more in the history will receive higher prediction scores, even though their items have the same matching level. Figure 1(b) shows empirical evidence from the FM on ML-1M dataset: among the items with the same ratings (*e.g.,* ratings = 4), the ones in the majority group will receive higher prediction scores. Therefore, the items in the majority group, including those undesirable or low-quality ones (see an example in Figure 1(c)), could deprive the recommendation opportunities of the items in the minority group.

The key to addressing bias amplification lies in eliminating the spurious correlation in the recommender modeling. To achieve this goal, we need to push the conventional RS to go beyond modeling the conditional probability and embrace the causal modeling of user representation on the prediction score. We propose a novel *Deconfounded Recommender System* (DecRS), which explicitly models the causal relations during training, and leverages backdoor adjustment [28] to eliminate the impact of the confounder. However, the sample space of the confounder is huge, making the traditional implementation of backdoor adjustment infeasible. To this end, we derive an approximation of backdoor adjustment, which is universally applicable to most recommender models. Lastly, we propose a user-specific inference strategy to dynamically regulate the influence of backdoor adjustment based on the user status. We instantiate DecRS on two representative models: FM [32] and neural factorization machines (NFM) [16]. Extensive experiments over two benchmarks demonstrate that our DecRS not only alleviates bias amplification effectively, but also improves the recommendation accuracy over the backbone models. The code and data are released at https://github.com/WenjieWWJ/DecRS.

Overall, the main contributions of this work are threefold:

- We construct a causal graph to analyze the causal relations in recommender models, which reveals the cause of bias amplification from a causal view.
- We propose a novel DecRS with an approximation of backdoor adjustment to eliminate the impact of the confounder, which can be incorporated into existing recommender models to alleviate bias amplification.
- We instantiate DecRS on two representative recommender models and conduct extensive experiments on two benchmarks, which validate the effectiveness of our proposal.

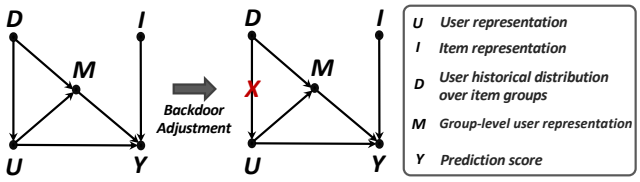


Figure 2: (a) The causal graph of conventional RS. (b) The causal graph used in DecRS.

2 METHODOLOGY

In this section, we first analyze the conventional RS from a causal view and explain the reason for bias amplification, which is followed by the introduction of the proposed DecRS.

2.1 A Causal View of Bias Amplification

To study bias amplification, we build up a causal graph to explicitly analyze the causal relations in the conventional RS.

2.1.1 Causal Graph. We scrutinize the causal relations in recommender models and abstract a causal graph, as shown in Figure 2(a), which consists of five variables: U, I, D, M, and Y. Note that we use the capital letter (*e.g.,* U), lowercase letter (*e.g.,* $\boldsymbol{u}$), and letter in the calligraphic font (*e.g.,* $\mathcal{U}$) to represent a variable, its particular value, and its sample space, respectively. In particular,

- U denotes user representation. For one user, $\boldsymbol{u} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_K]$ represents the embeddings of K user features (*e.g.,* ID, gender, and age) [32], where $\boldsymbol{u}_k \in \mathbb{R}^H$ is one feature embedding.
- I is item representation and each $\boldsymbol{i}$ denotes the embeddings of several item features (*e.g.,* ID and genre) which are similar to $\boldsymbol{u}$.
- D represents the user historical distribution over item groups. Groups can be decided by item attributes or similarity [35]. Given N item groups $\{g_1, ..., g_N\}$, $\boldsymbol{d}_u = [p_u(g_1), ..., p_u(g_N)] \in \mathbb{R}^N$ is a particular value of D when the user is u, where $p_u(g_n)$ is the click probability of user u over group g_n in the history[1]. For instance, for the user u in Figure 1(a), $\boldsymbol{d}_u$ is $[0.7, 0.3]$.
- M is the group-level user representation. A particular value $\boldsymbol{m} \in \mathbb{R}^H$ is a vector which describes how much the user likes different item groups. $\boldsymbol{m}$ can be obtained from the values of U and D. That is, M is deterministic if U and D are given so that we can represent $\boldsymbol{m}$ by a function $M(\boldsymbol{d}, \boldsymbol{u})$. We incorporate M into the causal graph because many recommender models (*e.g.,* FM) have modeled the user preference over item groups explicitly or implicitly by using the group-related features (*e.g.,* movie genre).
- Y with $y \in [0, 1]$ is the prediction score for the user-item pair.

The edges in the graph describe the causal relations between variables, *e.g.,* $U \rightarrow Y$ means that U has a direct *causal effect* [28] on Y, *i.e.,* changes on U will affect the value of Y. In particular,

- $D \rightarrow U$: the user historical distribution over item groups affects user representation U, making it favor the majority group. This is because user representation is optimized to fit the imbalanced historical data.

[1] In this work, we use click to represent any implicit feedback. For brevity, u and i denote the user and item, respectively. The click probability is obtained by normalizing the click frequency over groups.

- $(D, U) \rightarrow M$: D and U decide the group-level user representation.
- $(U, M, I) \rightarrow Y$: The edges show that U affects Y by two paths: 1) the direct path $U \rightarrow Y$, which denotes the user's pure preference over the item; and 2) the indirect path $U \rightarrow M \rightarrow Y$, indicating that the prediction score could be high because the user shows interest in the item group.

According to the causal theory [28], D is a *confounder* between U and Y, resulting in the spurious correlation.

2.1.2 Conventional RS. Due to the confounder, existing recommender models that estimate the conditional probability $P(Y|U, I)$ face the spurious correlation, which leads to bias amplification. Formally, given $U = \boldsymbol{u}$ and $I = \boldsymbol{i}$, we can derive the conditional probability $P(Y|U, I)$ by:

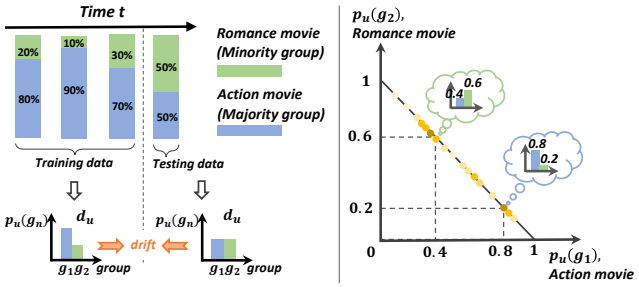
$$
\begin{aligned}
& P(Y|U=\boldsymbol{u}, I=\boldsymbol{i}) \\
&= \frac{\sum_{\boldsymbol{d}\in\mathcal{D}}\sum_{\boldsymbol{m}\in\mathcal{M}} P(\boldsymbol{d})P(\boldsymbol{u}|\boldsymbol{d})P(\boldsymbol{m}|\boldsymbol{d},\boldsymbol{u})P(\boldsymbol{i})P(Y|\boldsymbol{u},\boldsymbol{i},\boldsymbol{m})}{P(\boldsymbol{u})P(\boldsymbol{i})} && \text{(1a)} \\
&= \sum_{\boldsymbol{d}\in\mathcal{D}}\sum_{\boldsymbol{m}\in\mathcal{M}} P(\boldsymbol{d}|\boldsymbol{u})P(\boldsymbol{m}|\boldsymbol{d},\boldsymbol{u})P(Y|\boldsymbol{u},\boldsymbol{i},\boldsymbol{m}) && \text{(1b)} \\
&= \sum_{\boldsymbol{d}\in\mathcal{D}} P(\boldsymbol{d}|\boldsymbol{u})P(Y|\boldsymbol{u},\boldsymbol{i},M(\boldsymbol{d},\boldsymbol{u})) && \text{(1c)} \\
&= P(\boldsymbol{d}_u|\boldsymbol{u})P(Y|\boldsymbol{u},\boldsymbol{i},M(\boldsymbol{d}_u,\boldsymbol{u})), && \text{(1d)}
\end{aligned}
$$

where $\mathcal{D}$ and $\mathcal{M}$ are the sample spaces of D and M, respectively[2]. In particular, Eq. (1a) follows the law of total probability; Eq. (1b) is obtained by Bayes rule; since M can only take a value $M(\boldsymbol{d}, \boldsymbol{u})$, the sum over $\mathcal{M}$ in Eq. (1b) is removed, *i.e.,* $P(M(\boldsymbol{d}, \boldsymbol{u})|\boldsymbol{d}, \boldsymbol{u}) = 1$; and D is known if $U = \boldsymbol{u}$ is given. Thus $P(\boldsymbol{d}|\boldsymbol{u})$ is 1 if and only if $\boldsymbol{d}$ is $\boldsymbol{d}_u$; otherwise $P(\boldsymbol{d}|\boldsymbol{u}) = 0$, where $\boldsymbol{d}_u$ is the historical distribution of user u over item groups.

From Eq. (1d), we can find that $\boldsymbol{d}_u$ does not only affect the user representation $\boldsymbol{u}$ but also affects Y via $M(\boldsymbol{d}_u, \boldsymbol{u})$, causing the spurious correlation: given the item i in a group g_n, the more items in group g_n the user u has clicked in the history, the higher the prediction score Y becomes. In other words, the high prediction scores are caused by the users' historical interest in the group instead of the items themselves. From the perspective of model prediction, $\boldsymbol{d}_u$ affects $\boldsymbol{u}$, which makes $\boldsymbol{u}$ favor the majority group. In Eq. (1d), a higher click frequency $p_u(g_n)$ in $\boldsymbol{d}_u$ will make $M(\boldsymbol{d}_u, \boldsymbol{u})$ represent a strong interest in group g_n, increasing the prediction scores of items in group g_n via $P(Y|\boldsymbol{u}, \boldsymbol{i}, M(\boldsymbol{d}_u, \boldsymbol{u}))$. Consequently, the items in the majority group occupy the recommendation opportunities of items in the minority group, leading to bias amplification.

The spurious correlation is harmful for most users because the items in the majority group are likely to dominate the recommendation list and narrow down the user interest. Besides, the undesirable and low-quality items in the majority group will dissatisfy users, leading to poor recommendation accuracy. Worse still, by analyzing Eq. 1(d), we have a new observation: the prediction score Y heavily relies on the user historical distribution over item groups, *i.e.,* $\boldsymbol{d}_u$. Once $\boldsymbol{d}_u$ exhibits drift along time (see 3(a)), the recommendations will be dissatisfying.

[2]Theoretically, D has an infinite sample space. But the values are finite in a specific dataset. To simplify the notations, we use the discrete set $\mathcal{D}$ to represent the sample space of D, and so is M.



(a) User interest is changing over time. (b) Possible values of D and the probabilities.

Figure 3: (a) Illustration of user interest drift. (b) An example of the distribution of D where the item group number is 2. Each node in the line represents a particular value $\boldsymbol{d}$, and a darker color denotes a higher probability of $\boldsymbol{d}$, *i.e.,* $P(\boldsymbol{d})$.

2.2 Deconfounded Recommender System

To resolve the impact of the confounder, DecRS estimates the causal effect of user representation on the prediction score. Experimentally, the target can be achieved by collecting intervened data where the user representation is forcibly adjusted to eliminate the impact of the confounder. However, such an experiment is too costly to achieve in large-scale and faces the risk of hurting user experience in practice. DecRS thus resorts to the causal technique: *backdoor adjustment* [28, 29, 44], which enables the estimation of causal effect from the observational data.

2.2.1 Backdoor Adjustment. According to the theory of backdoor adjustment [28], the target of DecRS is formulated as: $P(Y|do(U = \boldsymbol{u}), I = \boldsymbol{i})$ where $do(U = \boldsymbol{u})$ can be intuitively seen as cutting off the edge $D \rightarrow U$ in the causal graph and blocking the effect of D on U (*cf.* Figure 2(b)). We then derive the specific expression of backdoor adjustment. Formally,

$$
\begin{aligned}
& P(Y|do(U=\boldsymbol{u}), I=\boldsymbol{i}) \\
&= \sum_{\boldsymbol{d}\in\mathcal{D}} P(\boldsymbol{d}|do(U=\boldsymbol{u}))P(Y|do(U=\boldsymbol{u}),\boldsymbol{i},M(\boldsymbol{d},do(U=\boldsymbol{u}))) && \text{(2a)} \\
&= \sum_{\boldsymbol{d}\in\mathcal{D}} P(\boldsymbol{d})P(Y|do(U=\boldsymbol{u}),\boldsymbol{i},M(\boldsymbol{d},do(U=\boldsymbol{u}))) && \text{(2b)} \\
&= \sum_{\boldsymbol{d}\in\mathcal{D}} P(\boldsymbol{d})P(Y|\boldsymbol{u},\boldsymbol{i},M(\boldsymbol{d},\boldsymbol{u})), && \text{(2c)}
\end{aligned}
$$

where the derivation of Eq. (2a) is the same as Eq. (1c), which follows the law of total probability and Bayes rule. Besides, Eq. (2b) and Eq. (2c) are obtained by two *do* calculus rules: *insertion/deletion of actions* and *action/observation exchange* in Theorem 3.4.1 of [28].

As compared to Eq. 1(d), DecRS estimates the prediction score with consideration of every possible value of D subject to the prior $P(\boldsymbol{d})$, rather than the probability of $\boldsymbol{d}$ conditioned on $\boldsymbol{u}$. Therefore, the items in the majority group will not receive high prediction scores purely because of a high click probability in $\boldsymbol{d}_u$. It thus alleviates bias amplification.

Intuitively, as shown in Figure 3(b), D has extensive possible values in a specific dataset, *i.e.,* users have various historical distributions over item groups. In DecRS, the prediction score Y considers various possible values of D. As such, 1) DecRS removes the dependency on $\boldsymbol{d}_u$ in Eq. 1(d) and mitigates the spurious

correlation, and 2) theoretically, when user interest drift happens, DecRS can produce more robust and satisfying recommendations because the model has "seen" many different values of D during training and doesn't heavily depend on the unreliable distribution $\boldsymbol{d}_u$ in Eq. 1(d).

2.2.2 Backdoor Adjustment Approximation. Theoretically, the sample space of D is infinite, which makes the calculation of Eq. (2c) intractable. Therefore, it is essential to derive an efficient approximation of Eq. (2c).

• *Sampling of D.* To estimate the distribution of D, we sample users' historical distributions over item groups in the training data, which comprise a discrete set $\tilde{\mathcal{D}}$. Formally, given a user u, $\boldsymbol{d}_u = [p_u(g_1), ..., p_u(g_N)] \in \tilde{\mathcal{D}}$ and each click frequency $p_u(g_n)$ over group g_n is calculated by:

$$p_u(g_n) = \sum_{i \in \mathcal{I}} p(g_n|i)p(i|u) = \frac{\sum_{i \in \mathcal{H}_u} q^i_{g_n}}{|\mathcal{H}_u|}, \tag{3}$$

where $\mathcal{I}$ is the set of all items, $\mathcal{H}_u$ denotes the clicked item set by user u, and $q^i_{g_n}$ represents the probability of item i belonging to group g_n. For instance, $\boldsymbol{q}^i = [1, 0, 0]$ with $q^i_{g_1} = 1$ denotes that item i only belongs to the first group. In this work, we sample D according to the user-item interactions in the training data, and thus the probability $P(\boldsymbol{d}_u)$ of user u is obtained by $\frac{|\mathcal{H}_u|}{\sum_{v \in \mathcal{U}} \mathcal{H}_v}$ where $\mathcal{U}$ represents the user set. After that, we can estimate Eq. (2c) by:

$$\begin{aligned} P(Y|do(U = \boldsymbol{u}), I = \boldsymbol{i}) &\approx \sum_{\boldsymbol{d} \in \tilde{\mathcal{D}}} P(\boldsymbol{d}) P(Y|\boldsymbol{u}, \boldsymbol{i}, M(\boldsymbol{d}, \boldsymbol{u})) \\ &= \sum_{\boldsymbol{d} \in \tilde{\mathcal{D}}} P(\boldsymbol{d}) f(\boldsymbol{u}, \boldsymbol{i}, M(\boldsymbol{d}, \boldsymbol{u})), \end{aligned} \tag{4}$$

where each $\boldsymbol{d}$ is a distribution from one user, and we use a function $f(\cdot)$ (*e.g.,* FM [32]) to calculate the conditional probability $P(Y|\boldsymbol{u}, \boldsymbol{i}, M(\boldsymbol{d}, \boldsymbol{u}))$, similar to conventional recommender models.

• *Approximation of $\mathbb{E}_{\boldsymbol{d}}[f(\cdot)]$.* The expected value of function $f(\cdot)$ of $\boldsymbol{d}$ in Eq. 4 is hard to compute because we need to calculate the results of $f(\cdot)$ for each $\boldsymbol{d}$ and the possible values in $\tilde{\mathcal{D}}$ are extensive. A popular solution [1, 38] in statistics and machine learning theory is to make the approximation $\mathbb{E}_{\boldsymbol{d}}[f(\cdot)] \approx f(\boldsymbol{u}, \boldsymbol{i}, M(\mathbb{E}_{\boldsymbol{d}}[\boldsymbol{d}], \boldsymbol{u}))$. Formally, the approximation takes the outer sum $\sum_{\boldsymbol{d}} P(\boldsymbol{d}) f(\cdot)$ into the calculation within $f(\cdot)$:

$$P(Y|do(U = \boldsymbol{u}), I = \boldsymbol{i}) \approx f(\boldsymbol{u}, \boldsymbol{i}, M(\sum_{\boldsymbol{d} \in \tilde{\mathcal{D}}} P(\boldsymbol{d})\boldsymbol{d}, \boldsymbol{u})). \tag{5}$$

The error of the approximation ϵ is measured by the *Jensen gap* [1]:

$$\epsilon = |\mathbb{E}_{\boldsymbol{d}}[f(\cdot)] - f(\boldsymbol{u}, \boldsymbol{i}, M(\mathbb{E}_{\boldsymbol{d}}[\boldsymbol{d}], \boldsymbol{u}))|. \tag{6}$$

Theorem 2.1. *If f is a linear function with a random variable X as the input, then $E[f(X)] = f(E[X])$ holds under any probability distribution $P(X)$. Refer to [1, 13] for the proof.*

Theorem 2.2. *If a random variable X with the probability distribution $P(X)$ has the expectation μ, and the non-linear function $f : G \to \mathbb{R}$ where G is a closed subset of $\mathbb{R}$, following:*

(1) f is bounded on any compact subset of G;

(2) $|f(x) - f(\mu)| = O(|x - \mu|^\beta)$ at $x \to \mu$ for $\beta > 0$;

(3) $|f(x)| = O(|x|^\gamma)$ as $x \to +\infty$ for $\gamma \geq \beta$,

then the inequality holds: $|\mathbb{E}[f(X)] - f(\mu)| \leq T(\rho^\beta_\beta + \rho^\gamma_\gamma)$, where $\rho_\beta = \sqrt[\beta]{\mathbb{E}[|X - \mu|^\beta]}$, and $T = sup_{x \in G \setminus \{\mu\}} \frac{|f(x) - f(\mu)|}{|x-\mu|^\beta + |x-\mu|^\gamma}$ does not depend on $P(X)$. The proof can be found in [13].

Table 1: Key notations and descriptions.

Notation	Description
$\boldsymbol{u} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_K], \boldsymbol{u}_k \in \mathbb{R}^H$	The representation vectors of K user features.
$\boldsymbol{x}_u = [x_{u,1}, ..., x_{u,K}]$	The feature values of a user's K features [32], *e.g.,* $[0.5, 1, ..., 0.2]$.
$\boldsymbol{d}_u = [p_u(g_1), ..., p_u(g_N)]$	$p_u(g_n)$ denotes the click frequency of user u over group g_n in the history, *e.g.,* $\boldsymbol{d}_u = [0.8, 0.2]$.
$\boldsymbol{m} = M(\boldsymbol{d}, \boldsymbol{u}) \in \mathbb{R}^H$	The group-level representation of user u under a historical distribution $\boldsymbol{d}$.
$\mathcal{H}_u$	The set of the items clicked by user u.
$\mathcal{U}, \mathcal{I}$	The user and item sets, respectively.
$\boldsymbol{q}^i = [q^i_{g_1}, ..., q^i_{g_N}] \in \mathbb{R}^N$	$q^i_{g_n}$ denotes the probability of item i belonging to group g_n, *e.g.,* $\boldsymbol{q}^i = [1, 0, 0]$.
$\boldsymbol{v} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_N], \boldsymbol{v}_n \in \mathbb{R}^H$	$\boldsymbol{v}_n$ denotes the representation of group g_n.
$\eta_u, \hat{\eta}_u$	The symmetric KL divergence value of user u and the normalized one, respectively.

From Theorem 2.1, we know that the error ϵ in Eq. 6 is zero if $f(\cdot)$ in Eq. 5 is a linear function. However, most existing recommender models use non-linear functions to increase the representation capacity. In these cases, there is an upper bound of ϵ which can be estimated by Theorem 2.2. It can be proven that the common non-linear functions in recommender models satisfy the conditions in Theorem 2.2, and the upper bound is small, especially when the distribution of D concentrates around its expectation [13].

2.3 Backdoor Adjustment Operator

To facilitate the usage of DecRS, we design the operator to instantiate backdoor adjustment, which can be easily plugged into recommender models to alleviate bias amplification. From Eq. 5, we can find that in addition to $\boldsymbol{u}$ and $\boldsymbol{i}$, $f(\cdot)$ takes $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ as the model input where $\bar{\boldsymbol{d}} = \sum_{\boldsymbol{d} \in \tilde{\mathcal{D}}} P(\boldsymbol{d})\boldsymbol{d}$. That is, if we can implement $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$, existing recommender models can take it as one additional input to achieve backdoor adjustment.

Recall that M denotes the group-level user representation which describes the user preference over item groups. Given $\bar{\boldsymbol{d}} = [p(g_1), ..., p(g_N)]$, item group representation $\boldsymbol{v} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_N]$, and user representation $\boldsymbol{u} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_K]$ with feature values $\boldsymbol{x}_u = [x_{u,1}, ..., x_{u,K}]$ [16], we calculate $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ by:

$$M(\bar{\boldsymbol{d}}, \boldsymbol{u}) = \sum_{a=1}^{N} \sum_{b=1}^{K} p(g_a)\boldsymbol{v}_a \odot x_{u,b}\boldsymbol{u}_b, \tag{7}$$

where $\odot$ denotes the element-wise product, and $\boldsymbol{v}_a \in \mathbb{R}^H$ is the item group representation for group g_a proposed by us, which is randomly initialized like $\boldsymbol{u}$. The feature values in $\boldsymbol{x}_u$ are usually one, but in some special cases, it could be a float number. For instance, a user may have two jobs and the feature value for these two features can be set as 0.5 separately. Besides, we can also leverage a FM module [32] or other high-order operators [10]. Formally, we can obtain $\boldsymbol{w} = [\bar{\boldsymbol{d}}, \boldsymbol{x}_u] = [p(g_1), ..., p(g_N), x_{u,1}, ..., x_{u,K}]$ and $\boldsymbol{c} = [\boldsymbol{v}, \boldsymbol{u}] = [\boldsymbol{v}_1, ..., \boldsymbol{v}_N, \boldsymbol{u}_1, ..., \boldsymbol{u}_K]$ via concatenation, and then

$M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ can be calculated by a second-order FM module:

$$M(\bar{\boldsymbol{d}}, \boldsymbol{u}) = \sum_{a=1}^{N+K} \sum_{b=1}^{N+K} w_a \boldsymbol{c}_a \odot w_b \boldsymbol{c}_b, \tag{8}$$

where $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ considers the interactions within $\boldsymbol{u}$ and $\boldsymbol{v}$ like FM, which is the main difference from Eq. 7.

Next, the group-level user representation $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ can be incorporated into existing recommender models as one additional user representation. Formally, if the generalized recommender models (*e.g.,* FM) are able to incorporate multiple feature representations, $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ is directly fed into the models to calculate $f(\boldsymbol{u}, \boldsymbol{i}, M(\bar{\boldsymbol{d}}, \boldsymbol{u}))$. Otherwise, $f(\cdot)$ can be implemented by a later-fusion manner [40], *i.e.,* $f(\cdot) = \delta * f^{'}(\boldsymbol{u}, \boldsymbol{i}) + (1-\delta) * f^{'}(M(\bar{\boldsymbol{d}}, \boldsymbol{u}), \boldsymbol{i})$ where δ is a hyperparameter and $f^{'}(\cdot)$ denotes the interaction module (*e.g.,* dot product) in recommender models to calculate the prediction score given user/item representations, such as neural collaborative filtering [17]. Then the parameters θ in the recommender models are optimized by:

$$\arg\min_{\theta} \sum_{(u,i,\bar{y}_{u,i}) \in \mathcal{T}} l\left(f(\boldsymbol{u}, \boldsymbol{i}, M(\bar{\boldsymbol{d}}, \boldsymbol{u})), \bar{y}_{u,i}\right), \tag{9}$$

where $\bar{y}_{u,i} \in \{0, 1\}$ represents whether user u has interacted with item i (*i.e.,* $\bar{y}_{u,i} = 1$) or not (*i.e.,* $\bar{y}_{u,i} = 0$), $\mathcal{T}$ denotes the training data, and $l(\cdot)$ is the loss function, *e.g.,* log loss [17].

2.4 Inference Strategy

As mentioned before, DecRS alleviates bias amplification and produces more robust predictions when user interest drift happens. Indeed, for some users, bias amplification might be beneficial to exclude the item groups they dislike. For example, users might only like action movies so that they don't watch the movies in other groups. In these special cases, it makes sense to purely recommend extensive action movies. Therefore, it is better to develop a user-specific inference strategy to regulate the impact of backdoor adjustment dynamically.

By analyzing the user behavior, we find that many users have diverse interest and are likely to have interest drift while few users have stable interest. This inspires us to explore the user characteristics: is this user easy to change the interest distribution over item groups? Based on that, we propose a user-specific inference strategy for item ranking. If the user is easy to change the interest distribution over item groups in the history, we assume that he/she has diverse interest and will change it easily in future. And thus backdoor adjustment is essential to alleviate bias amplification. Otherwise, the impact of backdoor adjustment should be controlled.

• *Symmetric KL Divergence.* We employ the symmetric Kullback–Leibler (KL) divergence to quantify the user interest drift in the history. In detail, we divide the historical interaction sequence of user u into two parts according to the timestamps. For each part, we calculate the historical distribution over item groups by Eq. 3, obtaining $\boldsymbol{d}_u^1 = [p_u^1(g_1), ..., p_u^1(g_N)]$ and $\boldsymbol{d}_u^2 = [p_u^2(g_1), ..., p_u^2(g_N)]$. Then, the distance between these two distributions is measured by the symmetric KL divergence:

$$\begin{aligned} \eta_u &= KL(\boldsymbol{d}_u^1 | \boldsymbol{d}_u^2) + KL(\boldsymbol{d}_u^2 | \boldsymbol{d}_u^1) \\ &= \sum_{n=1}^{N} P_u^1(g_n) \log \frac{P_u^1(g_n)}{P_u^2(g_n)} + \sum_{n=1}^{N} P_u^2(g_n) \log \frac{P_u^2(g_n)}{P_u^1(g_n)}, \end{aligned} \tag{10}$$

where η_u denotes the distribution distance of user u. A higher η_u represents that the user is easier to change the interest distribution over item groups. Here, we only divide the historical interaction sequence into two parts to reduce the computation cost. More fine-grained division can be explored in future work if necessary.

Based on the signal of η_u, we utilize an inference strategy to adaptively fuse the prediction scores from the conventional RS and DecRS. Specifically, we first train the recommender model by $P(Y|U = \boldsymbol{u}, I = \boldsymbol{i})$ and $P(Y|do(U = \boldsymbol{u}), I = \boldsymbol{i})$, respectively, and their prediction scores are then automatically fused to regulate the impact of backdoor adjustment. Formally,

$$Y_{u,i} = (1 - \hat{\eta}_u) * Y_{u,i}^{RS} + \hat{\eta}_u * Y_{u,i}^{DE}, \tag{11}$$

where $Y_{u,i}$ is the inference score for user u and item i, $Y_{u,i}^{RS}$ and $Y_{u,i}^{DE}$ are the prediction scores from the conventional RS and DecRS, respectively. In particular, $\hat{\eta}_u$ is calculated by:

$$\hat{\eta}_u = \left(\frac{\eta_u - \eta_{min}}{\eta_{max} - \eta_{min}}\right)^{\alpha}, \tag{12}$$

where the normalized $\hat{\eta}_u \in [0, 1]$, η_{min} and η_{max} are the minimum and maximum symmetric KL divergence values across all users, respectively. Besides, $\alpha \in [0, +\infty)$ is a hyper-parameter to further control the weights of $Y_{u,i}^{RS}$ and $Y_{u,i}^{DE}$ by human intervention. Specifically, $\hat{\eta}_u$ becomes larger if $\alpha \to 0$ due to $\hat{\eta}_u \in [0, 1]$ which makes $Y_{u,i}$ favor $Y_{u,i}^{DE}$, and $\hat{\eta}_u$ decreases if $\alpha \to +\infty$.

From Eq. 11, we can find that the inference for the users with high $\hat{\eta}_u$ will rely more on $Y_{u,i}^{DE}$. That is, η_u automatically adjusts the balance between $Y_{u,i}^{RS}$ and $Y_{u,i}^{DE}$. Besides, we can regulate the impact of backdoor adjustment by tuning the hyper-parameter α in Eq. 12 for different datasets or recommender models. Theoretically, α is usually close to 0 because mitigating the spurious correlation improves the recommendation accuracy for most users.

To summarize, the proposed DecRS has three main differences from the conventional RS:

- DecRS models the causal effect $P(Y|do(U = \boldsymbol{u}), I = \boldsymbol{i})$ instead of the conditional probability $P(Y|U = \boldsymbol{u}, I = \boldsymbol{i})$.
- DecRS equips the recommender models with a backdoor adjustment operator (Eq. 8).
- DecRS makes recommendations with a user-specific inference strategy instead of the simple model prediction (*e.g.,* a forward propagation).

3 RELATED WORK

In this work, we explore how to alleviate bias amplification of recommender models by causal inference, which is highly related to fairness, diversity, and causal recommendation.

Negative Effect of Bias Amplification. Due to the existence of feedback loop [7], bias amplification will become increasingly serious. Consequently, it will result in many negative issues: 1) narrowing down the user interest gradually, which is similar

to the effect of *filter bubbles* [23]. Worse still, the issue might evolve into *echo chambers* [14], in which users' imbalanced interest is further reinforced by the repeated exposure to similar items; 2) low-quality items that users dislike might be recommended purely because they are in the majority group, which deprive the recommendation opportunities of other high-quality items, causing low recommendation accuracy and unfairness.

Fairness in Recommendation. With the increasing attention on the fairness of machine learning algorithms [20], many studies explore the definitions of fairness in recommendation and information retrieval [21, 27, 30]. Generally speaking, they have two categories: individual fairness and group fairness. Individual fairness denotes that similar individuals (*e.g.,* users or items) should receive similar treatments (*e.g.,* exposure or clicks), such as amortized equity of attention [3]. Besides, group fairness indicates that all groups are supposed to be treated fairly where individuals are divided into groups according to the protected attributes (*e.g.,* item category and user gender) [22]. The particular definitions span from discounted cumulative fairness [46], fairness of exposure [34], to multi-sided fairness [5].

Another representative direction in fairness to reduce bias amplification is calibrated recommendation [35]. It re-ranks the items to make the distribution of the recommended item groups follow the proportion in the browsing history. For example, if a user has watched 70% action movies and 30% romance movies, the recommendation list is expected to have the same proportion. Although the fairness-related studies, including calibrated recommendation, may alleviate bias amplification well, they are making the trade-off between ranking accuracy and fairness [22, 34, 35]. The reason possibly lies in that they neglect the true cause of bias amplification.

Diversity in Recommendation. Diversity is regarded as one essential direction to get users out of filter bubbles in the information filtering systems [35]. As to recommendation, diversity pursues the dissimilarity of the recommended items [8, 36], where similarity can be measured by many factors, such as item category and embeddings [6, 18]. However, most studies might recommend many dissatisfying items when making diverse recommendations. For example, the recommender model may trade off the accuracy to reduce the intra-list similarity by re-ranking [49].

Causal Recommendation. Causal inference has been widely used in many machine learning applications, spanning from computer vision [26, 37], natural language processing [11, 12], to information retrieval [4, 47]. In recommendation, most studies on causal inference [28] focus on debiasing various biases in user feedback, including position bias [19], clickbait issue [40], and popularity bias [48]. The most representative idea in the existing work is *Inverse Propensity Scoring (IPS)* [2, 31, 44], which first estimates the propensity score based on some assumptions, and then uses the inverse propensity score to re-weight the samples. For instance, Saito *et al.* estimated the exposure propensity for each user-item pair, and re-weighted the samples via IPS to solve the miss-not-at-random problem [33]. However, IPS methods heavily rely on the accurate propensity estimation, and usually suffer from the high propensity variance. Thus it is often followed by the propensity clipping technique [2, 33]. Another line of causal recommendation studies the effect of taking recommendations as treatments on user/system behaviors [50], which is totally different from our work because we focus on debiasing recommendation.

Table 2: The statistics of the datasets.

Dataset	#Users	#Items	#Interactions	#Features	#Group
ML-1M	3,883	6,040	575,276	13,408	18
Amazon-Book	29,115	16,845	1,712,409	46,213	253

4 EXPERIMENTS

We conduct extensive experiments to demonstrate the effectiveness of our DecRS by investigating the following research questions:

- **RQ1:** How does the proposed DecRS perform across different users in terms of recommendation accuracy?
- **RQ2:** How does DecRS perform to alleviate bias amplification, compared to the state-of-the-art methods?
- **RQ3:** How do the different components affect the performance of DecRS, such as the inference strategy and the implementation of function $M(\cdot)$?

4.1 Experimental Settings

Datasets. We use two benchmark datasets, ML-1M and Amazon-Book, in different real-world scenarios. 1) ML-1M is a movie recommendation dataset[3], which involves rich user/item features, such as user gender, and movie genre. We partition the items into groups according to the movie genre. 2) Amazon-Book is one of the Amazon product datasets[4], where the book items can be divided into groups based on the book category (*e.g.,* sports). To ensure data quality, we adopt the 20-core settings, *i.e.,* discarding the users and items with less than 20 interactions. We summarize the statistics of datasets in Table 2.

For each dataset, we sort the user-item interactions by the timestamps, and split them into the training, validation, and testing subsets with the ratio of 80%, 10%, and 10%. For each interaction with the rating ≥ 4, we treat it as a positive instance. During training, we adopt the negative sampling strategy to randomly sample one item that the user did not interact with before as a negative instance.

Baselines. As our proposed DecRS is model-agnostic, we instantiate it on two representative recommender models, FM [32] and NFM [16], to alleviate bias amplification and boost the predictive performance. We compare DecRS with the state-of-the-art methods that might alleviate bias amplification of FM and NFM backbone models. In particular,

- **Unawareness** [15, 20] removes the features of item groups (*e.g.,* movie genre in ML-1M) from the input of item representation I.
- **FairCo** [22] introduces one error term to control the exposure fairness across item groups. In this work, we calculate the error term based on the ranking list sorted by relevance, and its coefficient λ in the ranking target is tuned in $\{0.01, 0.02, ..., 0.5\}$.
- **Calibration** [35] is one state-of-the-art method to alleviate bias amplification. Specifically, it proposes a calibration metric C_{KL} to measure the imbalance between the history and recommendation

[3] https://grouplens.org/datasets/movielens/1m/.
[4] https://jmcauley.ucsd.edu/data/amazon/.

Table 3: Overall performance comparison between DecRS and the baselines on ML-1M and Amazon-Book. %improv. denotes the relative performance improvement achieved by DecRS over FM or NFM. The best results are highlighted in bold.

Method	FM								NFM							
	ML-1M				Amazon-Book				ML-1M				Amazon-Book			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
FM/NFM [16, 32]	0.0676	0.1162	0.0566	0.0715	0.0213	0.0370	0.0134	0.0187	0.0659	0.1135	0.0551	0.0697	0.0222	0.0389	0.0144	0.0199
Unawareness [15]	0.0679	0.1179	0.0575	0.0730	0.0216	0.0377	0.0138	0.0191	0.0648	0.1143	0.0556	0.0708	0.0206	0.0381	0.0133	0.0190
FairCo [22]	0.0676	0.1165	0.0570	0.0720	0.0212	0.0370	0.0135	0.0188	0.0651	0.1152	0.0554	0.0708	0.0219	0.0390	0.0142	0.0199
Calibration [35]	0.0647	0.1149	0.0539	0.0695	0.0202	0.0359	0.0129	0.0181	0.0636	0.1131	0.0526	0.0682	0.0194	0.0335	0.0131	0.0178
Diversity [49]	0.0670	0.1159	0.0555	0.0706	0.0207	0.0369	0.0131	0.0185	0.0641	0.1133	0.0540	0.0693	0.0215	0.0386	0.0140	0.0197
IPS [33]	0.0663	0.1188	0.0556	0.0718	0.0213	0.0369	0.0135	0.0187	0.0648	0.1135	0.0544	0.0692	0.0213	0.0370	0.0137	0.0189
DecRS	**0.0704**	**0.1231**	**0.0578**	**0.0737**	**0.0231**	**0.0405**	**0.0148**	**0.0205**	**0.0694**	**0.1218**	**0.0580**	**0.0742**	**0.0236**	**0.0413**	**0.0153**	**0.0211**
%improv.	4.14%	5.94%	2.12%	3.08%	8.45%	9.46%	10.45%	9.63%	5.31%	7.31%	5.26%	6.46%	6.31%	6.17%	6.25%	6.03%

Table 4: Performance comparison across different user groups on ML-1M and Amazon-Book. Each line denotes the performance over the user group with $\eta_u >$ the threshold. We omit the results of threshold > 4 due to the similar trend.

	ML-1M						Amazon-Book					
FM	R@20			N@20			R@20			N@20		
Threshold	**FM**	**DecRS**	**%improv.**	**FM**	**DecRS**	**%improv.**	**FM**	**DecRS**	**%improv.**	**FM**	**DecRS**	**%improv.**
0	0.1162	0.1231	5.94%	0.0715	0.0737	3.08%	0.0370	0.0405	9.46%	0.0187	0.0205	9.63%
0.5	0.1215	0.1296	6.67%	0.0704	0.0730	3.69%	0.0383	0.0424	10.70%	0.0192	0.0213	10.94%
1	0.1303	0.1412	8.37%	0.0707	0.0741	4.81%	0.0430	0.0479	11.40%	0.0208	0.0232	11.54%
2	0.1432	0.1646	14.94%	0.0706	0.0786	11.33%	0.0518	0.0595	14.86%	0.0231	0.0274	18.61%
3	0.1477	0.1637	10.83%	0.0620	0.0711	14.68%	0.0586	0.0684	16.72%	0.0256	0.0318	24.22%
4	0.1454	0.1768	21.60%	0.0595	0.0737	23.87%	0.0659	0.0793	20.33%	0.0284	0.0362	27.46%
NFM	R@20			N@20			R@20			N@20		
Threshold	**NFM**	**DecRS**	**%improv.**	**NFM**	**DecRS**	**%improv.**	**NFM**	**DecRS**	**%improv.**	**NFM**	**DecRS**	**%improv.**
0	0.1135	0.1218	7.31%	0.0697	0.0742	6.46%	0.0389	0.0413	6.17%	0.0199	0.0211	6.03%
0.5	0.1187	0.1280	7.83%	0.0688	0.0735	6.83%	0.0401	0.0426	6.23%	0.0202	0.0218	7.92%
1	0.1272	0.1391	9.36%	0.0692	0.0747	7.95%	0.0438	0.0473	7.99%	0.0212	0.0234	10.38%
2	0.1452	0.1584	9.09%	0.0701	0.0771	9.99%	0.0530	0.0580	9.43%	0.0234	0.0269	14.96%
3	0.1478	0.1740	17.73%	0.0639	0.0723	13.15%	0.0614	0.0660	7.49%	0.0275	0.0319	16.00%
4	0.1442	0.1775	23.09%	0.0542	0.0699	28.97%	0.0709	0.0795	12.13%	0.0308	0.0371	20.45%

list, and minimizes C_{KL} by re-ranking. Here the hyper-parameter λ in the ranking target is searched in $\{0.01, 0.02, ..., 0.5\}$.

- **Diversity** [49] aims to decrease the intra-list similarity, where the diversification factor is tuned in $\{0.01, 0.02, ..., 0.2\}$.
- **IPS** [33] is a classical method in causal recommendation. Here we use $P(\boldsymbol{d}_u)$ as the propensity of user u to down-weight the items in the majority group during debiasing training, and we employ the propensity clipping technique [33] to reduce propensity variance, where the clipping threshold is searched in $\{2, 3, ..., 10\}$.

Evaluation Metrics. We evaluate the performance of all methods from two perspectives: recommendation accuracy and effectiveness of alleviating bias amplification. In terms of accuracy, two widely-used metrics [24, 43], Recall@K (R@K) and NDCG@K (N@K), are adopted under all ranking protocol [39, 42], which test the top-K recommendations over all items that users never interact with in the training data. As to alleviating bias amplification, we use the representative calibration metric C_{KL} [35], which quantifies the distribution drift over item groups between the history and the new recommendation list (comprised by the top-20 items). Higher C_{KL} scores suggest a more serious issue of bias amplification.

Parameter Settings. We implement our DecRS in the PyTorch implementation of FM and NFM. Closely following the original papers [16, 32], we use the following settings: in FM and NFM, the embedding size of user/item features is 64, log loss [17] is applied and the optimizer is set as Adagrad [9]; in NFM, a 64-dimension fully-connected layer is used. We adopt a grid search to tune their hyperparameters: the learning rate is searched in $\{0.005, 0.01, 0.05\}$; the batch size is tuned in $\{512, 1024, 2048\}$; the normalization coefficient is searched in $\{0, 0.1, 0.2\}$, and the dropout ratio is confirmed in $\{0.2, 0.3, ..., 0.5\}$. Besides, α in the proposed inference strategy is tuned in $\{0.1, 0.2, ..., 10\}$, and the model performs the best in $\{0.2, 0.3, 0.4\}$, where α is close to 0, proving the advantages of our DecRS over the conventional RS as discussed in Section 2.4. We use Eq. 8 to implement $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ and the backbone models take $M(\bar{\boldsymbol{d}}, \boldsymbol{u})$ as one additional feature. The exploration of the late-fusion manner is left to future work because it is not our main contribution. Furthermore, we use the early stopping strategy [41, 45] — stop training if R@10 on the validation set does not increase for 10 successive epochs. For all approaches, we tune the hyper-parameters to choose the best models *w.r.t.* R@10 on the validation set, and report the results on the testing set.

4.2 Performance Comparison (RQ1 & RQ2)

4.2.1 ***Overall Performance w.r.t. Accuracy.*** We present the empirical results of all baselines and DecRS in Table 3. Moreover, to further analyze the characteristics of DecRS, we split users into groups based on the symmetric KL divergence (*cf.* Eq. 10) and report

the performance comparison over the user groups in Table 4. From the two tables, we have the following findings:

- Unawareness and FairCo only achieve comparable performance or marginal improvements over the vanilla FM and NFM on the two datasets. Possible reasons are the trade-offs among different user groups. To be more specific, for some users, discarding group features or preserving group fairness is able to reduce bias amplification and recommend more satisfying items. However, for most users with imbalanced interest in item groups, these approaches possibly recommend many disappointing items by pursuing group fairness.
- Calibration and Diversity perform worse than the vanilla backbone models, suggesting that simple re-ranking does hurt the recommendation accuracy. This is consistent with the findings in [35, 49]. Moreover, we ascribe the inferior performance of IPS to the inaccurate estimation and high variance of propensity scores. That is, the propensity cannot precisely estimate the effect of D on U, even if the propensity clipping technique [33] is applied.
- DecRS effectively improves the recommendation performance of FM and NFM on the two datasets. As shown in Table 3, the relative improvements of DecRS over FM *w.r.t.* R@20 are 5.94% and 9.46% on ML-1M and Amazon-Book, respectively. This verifies the effectiveness of backdoor adjustment, which enables DecRS to remove the effect of confounder for many users. As a result, many less-interested or low-quality items from the majority group will not be recommended, thus increasing the accuracy.
- As Table 4 shows, with the increase of η_u, the performance gap between DecRS and the backbone models becomes larger. For example, in the user group with $\eta_u > 4$, the relative improvements *w.r.t.* N@20 over FM and NFM are 23.87% and 28.97%, respectively. We attribute such improvements to the robust recommendation produced by DecRS. Specifically, DecRS equipped with backdoor adjustment is superior in reducing the spurious correlation and predicting users' diverse interest, especially for the users with the interest drift (*i.e.,* high η_u).

4.2.2 ***Performance on Alleviating Bias Amplification.*** In Figure 4, we present the performance comparison *w.r.t.* C_{KL} between the vanilla FM/NFM, calibrated recommendation, and DecRS on ML-1M. Due to space limitation, we omit other baselines that perform worse than calibrated recommendation and the results on Amazon-Book which have similar trends. We have the following observations from Figure 4. 1) As compared to the vanilla models, calibrated recommendation achieves lower C_{KL} scores, suggesting that the bias amplification is reduced. However, it comes at the cost of lower recommendation accuracy, as shown in Table 3. 2) Our DecRS consistently achieves lower C_{KL} scores than calibrated recommendation across all user groups. More importantly, DecRS does not hurt the recommendation accuracy. This evidently shows that DecRS solves the bias amplification problem well by embracing causal modeling for recommendation, and justifies the effectiveness of backdoor adjustment on reducing spurious correlations.

4.3 In-depth Analysis (RQ3)

4.3.1 ***Effect of the Inference Strategy.*** We first answer the question: is it of importance to conduct the inference strategy for DecRS? Towards this end, one variant "DecRS (w/o)" is constructed by disabling the inference strategy and only using the prediction Y^{DE} in Eq. 11 for inference. We illustrate its results in Figure 5 with the following key findings. 1) The performance of "DecRS (w/o)" drops as compared with that of DecRS, indicating the effectiveness of the inference strategy. 2) "DecRS (w/o)" still outperforms FM and NFM consistently, especially over the users with high η_u. This suggests the superiority of DecRS over the conventional RS. It achieves more accurate predictions of user interest by mitigating the effect of the confounder via backdoor adjustment approximation.

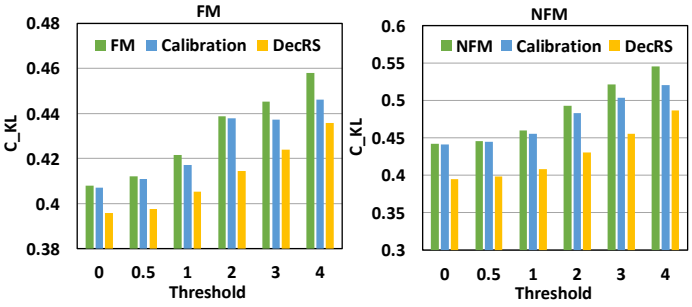


Figure 4: The performance comparison between the baselines and DecRS on alleviating bias amplification.

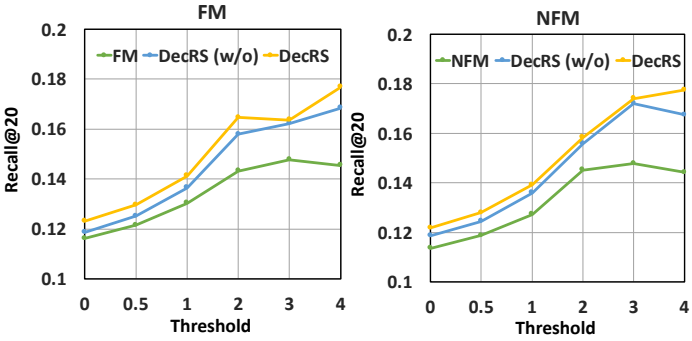


Figure 5: Ablation study of DecRS on ML-1M.

Table 5: Effect of the design of $M(\cdot)$.

Method	R@10	R@20	N@10	N@20
FM	0.0676	0.1162	0.0566	0.0715
DecRS-EP	0.0685	0.1205	0.0573	0.0730
DecRS-FM	0.0704	0.1231	0.0578	0.0737

4.3.2 ***Effect of the Implementation of*** $M(\cdot)$. As mentioned in Section 2.3, we can implement the function $M(\cdot)$ by either Eq. 7 or Eq. 8. We investigate the influence of different implementations and construct two variants, DecRS-EP and DecRS-FM, which employ the element-wise product in Eq. 7 and the FM module in Eq. 8, respectively. We summarize their performance comparison over FM on ML-1M in Table 5. While being inferior to DecRS-FM, DecRS-EP still performs better than FM. This validates the superiority of DecRS-FM over DecRS-EP, and also shows that DecRS with different implementations still surpasses the vanilla backbone models, which further suggests the stability and effectiveness of DecRS.

5 CONCLUSION AND FUTURE WORK

In this work, we explained that bias amplification in recommender models is caused by the confounder. To alleviate bias amplification, we proposed a novel DecRS with an approximation operator for backdoor adjustment. DecRS explicitly models the causal relations in recommender models, and leverages backdoor adjustment to remove the spurious correlation caused by the confounder. Besides, we developed an inference strategy to regulate the impact of backdoor adjustment. Extensive experiments validate the effectiveness of DecRS on alleviating bias amplification and improving recommendation accuracy.

This work takes the initial step to incorporate backdoor adjustment into existing recommender models, which opens up many promising research directions. For instance, 1) the discovery of more fine-grained causal relations. Recommendation is a complex scenario, involving many observed/hidden variables, which can result in confounding. 2) DecRS has the potential to reduce various biases caused by the imbalanced training data, such as position bias and popularity bias. 3) Bias amplification is one essential cause of the filter bubble [23] and echo chambers [14]. The effect of DecRS on mitigating these issues can be studied in future work.

REFERENCES

[1] Shoshana Abramovich and Lars-Erik Persson. 2016. Some new estimates of the 'Jensen gap'. *Journal of Inequalities and Applications* 2016, 1 (2016), 1–9.

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W. Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *SIGIR*. ACM, 385–394.

[3] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*. ACM, 405–414.

[4] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *RecSys*. ACM, 104–112.

[5] Robin Burke. 2017. Multisided fairness for recommendation. In *FAT ML*.

[6] Praveen Chandar and Ben Carterette. 2013. Preference based evaluation measures for novelty and diversity. In *SIGIR*. ACM, 413–422.

[7] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *RecSys*. ACM, 224–232.

[8] Peizhe Cheng, Shuaiqiang Wang, Jun Ma, Jiankai Sun, and Hui Xiong. 2017. Learning to Recommend Accurate and Diverse Items. In *WWW*. IW3C2, 183–192.

[9] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* 12, 7 (2011).

[10] Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Cross-GCN: Enhancing Graph Convolutional Network with k-Order Feature Interactions. *TKDE* (2021).

[11] Fuli Feng, Weiran Huang, Xin Xin, Xiangnan He, and Tat-Seng Chua. 2021. Should Graph Convolution Trust Neighbors? A Simple Causal Inference Method. In *SIGIR*. ACM.

[12] Fuli Feng, Jizhi Zhang, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Empowering Language Understanding with Counterfactual Reasoning. In *ACL-IJCNLP Findings*. ACL.

[13] Xiang Gao, Meera Sitharam, and Adrian E. Roitberg. 2019. Bounds on the Jensen Gap, and Implications for Mean-Concentrated Distributions. *AJMAA* 16, 14 (2019), 1–16. Issue 2.

[14] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. 2020. Understanding Echo Chambers in E-Commerce Recommender Systems. In *SIGIR*. ACM, 2261–2270.

[15] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. 2016. The case for process fairness in learning: Feature selection for fair decision making. In *NeuIPS*.

[16] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. ACM, 355–364.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.

[18] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. 2020. What Aspect Do You Like: Multi-Scale Time-Aware User Interest Modeling for Micro-Video Recommendation. In *MM*. ACM, 3487–3495.

[19] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM*. ACM, 781–789.

[20] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *NeuIPS*. Curran Associates, Inc., 4066–4076.

[21] 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 and satisfaction in recommendation systems. In *CIKM*. ACM, 2243–2251.

[22] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. In *SIGIR*. ACM, 429–438.

[23] 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 *WWW*. ACM, 677–686.

[24] Liqiang Nie, Yongqi Li, Fuli Feng, Xuemeng Song, Meng Wang, and Yinglong Wang. 2020. Large-Scale Question Tagging via Joint Question-Topic Embedding Learning. *TOIS* 38 (2020).

[25] Liqiang Nie, Meng Liu, and Xuemeng Song. 2019. Multimodal learning toward micro-video understanding. *Synthesis Lectures on Image, Video, and Multimedia Processing* 9, 4 (2019), 1–186.

[26] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *CVPR*. IEEE.

[27] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2020. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *WWW*. ACM, 1194–1204.

[28] Judea Pearl. 2009. *Causality*. Cambridge university press.

[29] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect* (1st ed.). Basic Books, Inc.

[30] Evaggelia Pitoura, Georgia Koutrika, and Kostas Stefanidis. 2020. Fairness in Rankings and Recommenders.. In *EDBT*. ACM, 651–654.

[31] Zhen Qin, Suming J. Chen, Donald Metzler, Yongwoo Noh, Jingzheng Qin, and Xuanhui Wang. 2020. Attribute-Based Propensity for Unbiased Learning in Recommender Systems: Algorithm and Case Studies. In *KDD*. ACM, 2359–2367.

[32] Steffen Rendle. 2010. Factorization machines. In *ICDM*. IEEE, 995–1000.

[33] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *WSDM*. ACM, 501–509.

[34] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *KDD*. ACM, 2219–2228.

[35] Harald Steck. 2018. Calibrated recommendations. In *RecSys*. ACM, 154–162.

[36] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, and Mark Coates. 2020. A Framework for Recommending Accurate and Diverse Items Using Bayesian Graph Convolutional Neural Networks. In *KDD*. ACM, 2030–2039.

[37] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-Tailed Classification by Keeping the Good and Removing the Bad Momentum Causal Effect. In *NeuIPS*.

[38] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. 2020. Visual commonsense r-cnn. In *CVPR*. IEEE, 10760–10770.

[39] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *WSDM*. ACM, 373–381.

[40] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Click can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *SIGIR*. ACM.

[41] Wenjie Wang, Minlie Huang, Xin-Shun Xu, Fumin Shen, and Liqiang Nie. 2018. Chat more: Deepening and widening the chatting topic via a deep model. In *SIGIR*. ACM, 255–264.

[42] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. ACM, 165–174.

[43] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. ACM, 1001–1010.

[44] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2018. The deconfounded recommender: A causal inference approach to recommendation. In *arXiv:1808.06581*.

[45] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *MM*. ACM, 1437–1445.

[46] Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *SSDBM*. ACM, 1–6.

[47] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. 2021. Deconfounded Video Moment Retrieval with Causal Intervention. In *SIGIR*. ACM.

[48] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR*. ACM.

[49] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *WWW*. ACM, 22–32.

[50] Hao Zou, Peng Cui, Bo Li, Zheyan Shen, Jianxin Ma, Hongxia Yang, and Yue He. 2020. Counterfactual Prediction for Bundle Treatment. *NeuIPS* (2020).