ReCRec: Reasoning the Causes of Implicit Feedback for Debiased Recommendation

SIYI LIN, College of Computer Science, Zhejiang University, Hangzhou, China
SHENG ZHOU, School of Software Technology, Zhejiang University, Hangzhou, China
JIAWEI CHEN, State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, China
YAN FENG, State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and College of Computer Science, Zhejiang University, Hangzhou, China
QIHAO SHI, Zhejiang University, Hangzhou, China
CHUN CHEN, State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and College of Computer Science, Zhejiang University, Hangzhou, China
YING LI, Bangsun Technology, Hangzhou, China
CAN WANG, State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, China

Implicit feedback (e.g., user clicks) is widely used in building recommender systems (RS). However, the inherent notorious *exposure bias* significantly affects recommendation performance. Exposure bias refers a phenomenon that implicit feedback is influenced by user exposure and does not precisely reflect user preference. Current methods for addressing exposure bias primarily reduce confidence in unclicked data, employ exposure models, or leverage propensity scores. Regrettably, these approaches often lead to biased estimations or elevated model variance, yielding sub-optimal results.

To overcome these limitations, we propose a new method **ReCRec** that **Reasons the *C*auses behind the implicit feedback for debiased R*e*commendation**. ReCRec identifies three scenarios behind unclicked data—i.e., unexposed, dislike, or a combination of both. A reasoning module is employed to infer the category to which each instance pertains. Consequently, the model is capable of extracting reliable positive and negative

This work was supported by the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-001), the National Natural Science Foundation of China (62372399), and the advanced computing resources provided by the Supercomputing Center of Hangzhou City University.
Authors' Contact Information: Siyi Lin, College of Computer Science, Zhejiang University, Hangzhou, China; e-mail: lin452_lsy@zju.edu.cn; Sheng Zhou, School of Software Technology, Zhejiang University, Hangzhou, China; e-mail: zhousheng_zju@zju.edu.cn; Jiawei Chen (Corresponding author), State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, China; e-mail: sleepyhunt@zju.edu.cn; Yan Feng, State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and College of Computer Science, Zhejiang University, Hangzhou, China; e-mail: fengyan@zju.edu.cn; Qihao Shi, Zhejiang University, Hangzhou, China; e-mail: shiqihao321@zju.edu.cn; Chun Chen, State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and College of Computer Science, Zhejiang University, Hangzhou, China; e-mail: chenc@zju.edu.cn; Ying Li, Bangsun Technology, Hangzhou, China; e-mail: li_ying@bisfit.com.cn; Can Wang, State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China and Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, China; e-mail: wcan@zju.edu.cn.

ACM 1558-2868/2024/10-ART158
https://doi.org/10.1145/3672275

signals from unclicked data, thereby facilitating more accurate learning of user preferences. We also conduct thorough theoretical analyses to demonstrate the debiased nature and low variance of ReCRec. Extensive experiments on both semi-synthetic and real-world datasets validate its superiority over state-of-the-art methods.

CCS Concepts: • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Debias, recommendation, implicit feedback

ACM Reference format:
Siyi Lin, Sheng Zhou, Jiawei Chen, Yan Feng, Qihao Shi, Chun Chen, Ying Li, and Can Wang. 2024. ReCRec: Reasoning the Causes of Implicit Feedback for Debiased Recommendation. *ACM Trans. Inf. Syst.* 42, 6, Article 158 (October 2024), 26 pages.
https://doi.org/10.1145/3672275

1 Introduction

Being able to provide personalized suggestions, **Recommender System (RS)** has been widely applied in many online services [19, 67, 73]. Modern RS is usually built on the implicit feedback data[1] (e.g., user clicks), which are natural byproducts of user behavior and more abundant than the explicit counterpart (e.g., ratings) [50]. However, handling implicit feedback is more challenging due to the inherent notorious *exposure bias* [5]. Implicit feedback is a consequence of both preference and exposure, as opposed to purely signifying user preference. Blindly fitting data without tackling exposure bias will result in inferior performance.

Recent years have witnessed a multitude of studies on addressing exposure bias, which can generally be categorized into three types:

- **Weighted Matrix Factorization (WMF)** [28] simply considers all unclicked data as negative while downweighing their contributions. But in fact, unclicked data does not always mean a user dislikes an item. It may be simply because the user is unexposed to the item, while he is actually fond of it instead.
- **Exposure-based Models (Exposure MF (ExpoMF))** [38] deduce how likely a user is exposed to an item, and learn a recommendation model on the *exposed data* (i.e., the data where an item is exposed). However, since items are not always evenly exposed in practical, the exposed data may show skewed patterns of user preference, e.g., a tendency toward popular items. Bias issue remains unresolved for ExpoMF.
- **Propensity-based Models (Inverse Probability Weighting (IPW))** [34, 35, 48, 52, 53, 76] treat unclicked data as negative as usual but leverage a subtle propensity strategy to offset its negative effect. Despite theoretical unbiasedness, this strategy incurs high model variance. Additionally, finding the proper propensity is itself a challenging problem.

In summary, we argue that existing methods do not fully exploit unclicked data. In fact, as shown in Figure 1(a), unclicked data are a mixture of the following three cases: (1) a user likes but does not know an item (positive); (2) a user knows but dislikes an item (negative); (3) a user neither likes nor knows an item (negative). *Both positive and negative signals are contained in unclicked data, while positive signals are always overlooked by existing methods*—they either discard positive cases (e.g., ExpoMF) or erroneously treat positive as negative (e.g., WMF, IPW). To overcome this

[1]In this work, we simply use the term "click" as a placeholder of implicit feedback for better description, while it can be replaced by other forms of implicit feedback.

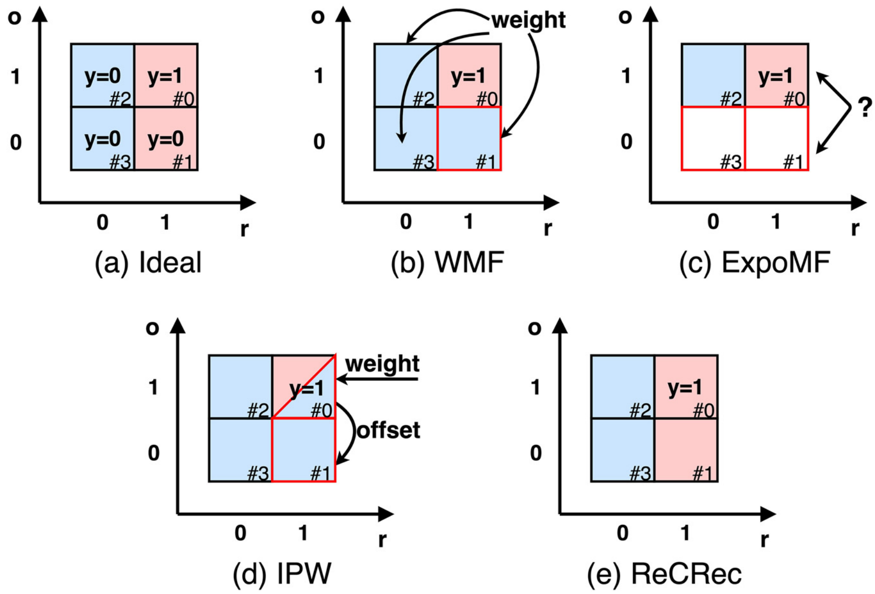


Fig. 1. Illustrating how ideal loss, WMF, ExpoMF, IPW, and ReCRec treat four cases, where $y \in \{0, 1\}$ denotes the implicit feedback, and $o \in \{0, 1\}$ (or $r \in \{0, 1\}$) indicates whether a user is exposed to (or fond of) an item. The grid in red, blue, or white reflects the case is treated as positive, negative, or be overlooked, respectively. The red line denotes cases that have been incorrectly handled by the method.

problem and for extracting reliable positive and negative signals, it is essential to deduce the cases that each instance belongs to.

Toward this end, this work proposes a new method named **ReCRec** that **Reasons the Causes behind implicit feedback for unbiased Recommendation**. ReCRec contains three modules: (1) **Preference Module (PM)** for capturing user preference; (2) **Exposure Module (EM)** for inferring user exposure; and (3) **Reasoning Module (RM)** targeting at reasoning to which case each instance belongs. An **Expectation-Maximization (E-M)** algorithm is developed to update the modules iteratively and alternatively. The learning of user preference and exposure can benefit from the reliable signals processed by the RM, while the RM can reversely utilize the knowledge from PM and EM. We further conduct theoretical analyses on ReCRec, proving that ReCRec achieves unbiased estimation and has relatively lower variance compared with IPW.

To summarize, this work makes the following contributions:

— Positioning unclicked data in three cases and revealing the intrinsic flaws of existing methods—they fail to extract reliable positive and negative signals from unclicked data.
— Proposing a new method ReCRec for addressing exposure bias, which explicitly reasons the causes behind implicit feedback with sound theoretical foundation.
— Conducting extensive experiments on both semi-synthetic and real-world datasets to demonstrate the superiority of ReCRec over state-of-the-art.

The remainder of this article is organized as follows: Section 2 provides the analyses of existing methods addressing exposure bias. The proposed method ReCRec is introduced in Section 4. Section 5 presents our experimental results and subsequent discussions. Related work is reviewed in Section 6. Finally, we conclude the article and present some future directions in Section 7.

Table 1. Notations in This Article

Notations	Descriptions
u	a user in the user set $\mathcal{U}$
i	an item in the item set $\mathcal{I}$
$\mathcal{D}$	a set of all user–item pairs
y_{ui}	whether the user u clicks the item i
r_{ui}	whether the user u is interested in the item i
o_{ui}	whether the user u is exposed to the item i
x_u, x_i	features of the user u and item i
ξ_{ui}	the ground-truth preference level for the user–item pair (u, i)
$\hat{\xi}_{ui}$	the model prediction on the preference for the pair (u, i)
θ_{ui}	the ground-truth exposure probability for the pair (u, i)
$\hat{\theta}_{ui}$	the model prediction on the exposure for the pair (u, i)
$\delta_{ui}^{(r)}$	the loss between the prediction $\hat{\xi}_{ui}$ and the label r
Φ_{ui}	predicted distribution over four cases for the user–item pair (u, i)
$\varphi_{ui}^{(ab)}$	predicted probability of each case for the pair (u, i), *i.e.*, $\varphi_{ui}^{(ab)} = \tilde{P}(r_{ui} = a, o_{ui} = b \mid y_{ui})$
$L_{\text{ideal}}(\hat{\xi})$	the ideal loss for training a recommendation (preference) model
$L_{\text{PM}}(\hat{\xi})$	the loss for training the PM in ReCRec
$L_{\text{EM}}(\hat{\theta})$	the loss for training the EM in ReCRec

2 Preliminaries

In this section, we first formulate the task of unbiased recommendation from implicit feedback and then briefly revisit existing methods on this problem. Table 1 summarizes the notations involved in this article.

2.1 Recommendation with Implicit Feedback

Suppose we have a RS with a user set $\mathcal{U}$ and an item set $\mathcal{I}$. Let $u \in \mathcal{U}$ (or $i \in \mathcal{I}$) be a user (or an item) and $\mathcal{D} = \mathcal{U} \times \mathcal{I}$ be a set of all user–item pairs. Users' historical interactions with items can be expressed as a matrix $\mathrm{Y} \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{I}|}$, whose element y_{ui} represents whether a user u has interacted with an item i ($y_{ui} = 1$) or not ($y_{ui} = 0$). There are numerous types of interactions in implicit feedback, such as clicks, purchases, and so on. For the sake of convenience, in this work, we simply use the term "click" as a typical representative for further description. In implicit recommendation, click (y_{ui}) is considered as the consequence of user exposure and preference. Here, we refer to recent work [53, 69] and introduce two latent Bernoulli variables for better description: (1) o_{ui} representing whether the user is exposed to the item and (2) r_{ui} representing whether the user is actually interested in the item. Existing studies [53] often make the following assumption:

$$y_{ui} = o_{ui} \cdot r_{ui}. \tag{1}$$

That is, a user clicks an item only when he is both aware and fond of the item. Equation (1) clearly reveals *exposure bias* inherent in the implicit feedback data—i.e., click does not fully represent like, as the absence of click may be due to lack of exposure rather than disinterest. The task of unbiased implicit recommendation can be stated as follows: learning a preference model from Y for predicting user–item preference accurately.

The ideal loss for training or evaluating a recommendation model is defined as follows[2] [53]:

$$L_{\text{ideal}}(\hat{\xi}) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \Big(\xi_{ui} \delta^{(1)}(\hat{\xi}_{ui}) + (1 - \xi_{ui})\, \delta^{(0)}(\hat{\xi}_{ui}) \Big), \tag{2}$$

where ξ_{ui} denotes the ground-truth preference level, characterizing how likely a user u likes an item i (i.e., $\xi_{ui} = P(r_{ui} = 1)$); $\hat{\xi}_{ui}$ denotes the prediction from the recommendation model; and $\delta^{(r)}(\hat{\xi}_{ui})$ measures the difference between the prediction ($\hat{\xi}_{ui}$) and the label ($r \in \{0, 1\}$). δ can be appropriately chosen for recovering most point-wise loss function, e.g., $\delta^{(r)}(\hat{\xi}_{ui}) = -r \log(\hat{\xi}_{ui}) - (1 - r) \log(1 - \hat{\xi}_{ui})$ for cross-entropy loss and $\delta^{(r)}(\hat{r}_{ui}) = (\hat{r}_{ui} - r)^2$ for RMSE loss. In this article, we usually simplify the notation $\delta^{(r)}(\hat{\xi}_{ui})$ as $\delta_{ui}^{(r)}$ for better presentation.

Since ξ in ideal loss is not available, the model training can be only conducted on the click data Y. Hence, the exploration for a suitable surrogate loss toward unbiased estimation of the ideal loss is ongoing.

2.2 Four Cases behind Implicit Feedback

To better understand the relation and gap between the click and the relevance, we identify four cases behind the implicit feedback according to the user exposure and preference. Specifically, when $y_{ui} = 1$, we have:

— *Case#0*: $r_{ui} = 1$ and $o_{ui} = 1$, the user both knows and likes the item.

When $y_{ui} = 0$, as shown in Figure 1, the unclicked data can be positioned as one of the following three cases:

— *Case#1*: $r_{ui} = 1$ and $o_{ui} = 0$, the user likes but does not know the item.
— *Case#2*: $r_{ui} = 0$ and $o_{ui} = 1$, the user knows but dislikes the item.
— *Case#3*: $r_{ui} = 0$ and $o_{ui} = 0$, the user is neither exposed to nor interested in the item.

This insight clearly reveals the inconsistence of the click and relevance. Specifically, unclicked data do not always mean negative signal. It may lies on Case#1 ($r_{ui} = 1, o_{ui} = 0$), where the user likes the item instead.

2.3 Analyses on Existing Methods

Based on the above insight, we now revisit existing methods and discuss their limitations. Existing methods are mainly three types:

(1) WMF [28] simply considers all unclicked data as negative while downweighing their contributions:

$$L_{\text{WMF}}(\hat{\xi}) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \Big(y_{ui} \delta_{ui}^{(1)} + (1 - y_{ui})\, w_{ui} \delta_{ui}^{(0)} \Big), \tag{3}$$

where a confidence weight w_{ui} for the unclicked data is introduced, which is usually heuristically specified with a uniform low value or based on item popularity. As Figure 1(b) shows, WMF blindly considers all unclicked data as negative and may erroneously flip the label of the user–item pairs that belong to Case#1 ($r_{ui} = 1, o_{ui} = 0$). As such, the loss of WMF is biased against the ideal one.

[2]Here, we simply take the representative point-wise loss for analyses as recent work on exposure bias [38, 53].

(2) Exposure-based models (ExpoMF) [38] infer how likely a user is exposed to an item ($\hat{P}(o_{ui})$) and optimize

$$L_{\text{ExpoMF}}(\hat{\xi}) = \frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} \mathbb{E}_{o_{ui}\sim\hat{P}(o_{ui})} \left[y_{ui}\delta_{ui}^{(1)} + (1-y_{ui})\, o_{ui}\delta_{ui}^{(0)} \right]. \tag{4}$$

The loss function only considers the scenarios in which the item is exposed to the user ($o_{ui} = 1$), i.e., the loss is computed exclusively when $y_{ui} = 1$ or $y_{ui} = 0$ & $o_{ui} = 1$. ExpoMF only takes Cases 0 and 2 to learn the user preference, as Figure 1(c) shows. This dedicated strategy can obviate mistaken labels, since when $o_{ui} = 1$ the equation $y_{ui} = r_{ui}$ holds. However, as items in practical are not always evenly exposed, the exposed data would exhibit skewed patterns of user preference. ExpoMF would be typically biased toward the items with higher exposure opportunity (e.g., popular items), reinforcing their performance, while discriminating the items with lower exposure probability (e.g., niche items). Bias issue remains unresolved for ExpoMF. This point has also been uncovered by recent work [53] with theoretical analyses. In fact, we have:

Lemma 1 (Bias of ExpoMF [53]). *ExpoMF cannot provide an unbiased estimation of the ideal loss when a subset of items are unexposed to a user (i.e., if* $\exists\, u \in U$ *&* $i \in I, \theta_{ui} \neq 1$*, we have* $\mathbb{E}[L_{\text{ExpoMF}}(\hat{\xi})] \neq L_{ideal}(\hat{\xi})$*).*

The proof procedure is delineated in Appendix A, referring to [53]. Considering that items cannot be fully exposed to users in practical scenarios, bias is commonly encountered in ExpoMF. This issue originates from its oversight of Cases 1 and 3, which potentially contain valuable positive and negative preference signals. Therefore, it is crucial to consider all four cases to learn user preferences effectively and without bias.

(3) Propensity-based models [53] treat unclicked data as negative while reweighing clicked data with the inverse of the propensity:

$$L_{\text{IPW}}(\hat{\xi}) = \frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} \left(y_{ui} \left(\frac{1}{\rho_{ui}} \delta_{ui}^{(1)} + \left(1 - \frac{1}{\rho_{ui}}\right) \delta_{ui}^{(0)} \right) + (1-y_{ui})\, \delta_{ui}^{(0)} \right), \tag{5}$$

where ρ_{ui} is defined as propensity, which estimates the probability of the exposure. L_{IPW} could theoretically achieve unbiased estimation with a proper propensity (i.e., $\rho_{ui} = P(o_{ui} = 1)$). Although this method has mis-specified the labels of the data belonging to Case#1, the subtle design of the propensity could properly offset this negative effect. However, we remark that this treatment would incur high variance with

$$\mathbb{V}[L_{\text{IPW}}] = \frac{1}{|\mathcal{D}|^2} \sum_{(u,i)\in\mathcal{D}} \xi_{ui} \left(\frac{1}{\rho_{ui}} - \xi_{u,i} \right) \left(\delta_{ui}^{(1)} - \delta_{ui}^{(0)} \right)^2. \tag{6}$$

The variance depends on the inverse of the propensity, which is potentially exploded especially for the items with low exposure probability. Although the clipping technique [10] could be applied to reduce the variance, the unbiasedness would also be broken.

Summary. Existing methods do not well exploit the unclicked data: ExpoMF only utilizes partial instances belonging to Case#0 and Case#2, ignoring Case#1 that signifies user positive preference; while IPW and WMF mistakenly treated some positive instances (Case#1) as negative. This intrinsic flaw heavily hinders the model from making accurate recommendation. To overcome this limitation

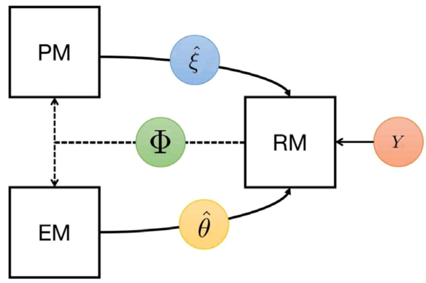


Fig. 2. Illustration of training process of ReCRec, where $\hat{\xi}$ and $\hat{\theta}$ denote the predicted preference probability and exposure probability, and Φ denotes the deduced distribution over four cases.

and for extracting reliable positive and negative signals, we need to explicitly deduce the cases behind the user implicit feedback.

3 Proposed Method: ReCRec

In this section, we first detail the proposed ReCRec and then conduct thorough theoretical analyses to show its merits.

3.1 Architecture of ReCRec

This work proposes a new method ReCRec that reasons the causes behind the implicit feedback for debiased recommendation. ReCRec includes the following three modules:

- PM aims at capturing user true preference. It can be implemented by various existing methods like MF [33] or LightGCN [21]. Formally, given the features (x_u, x_i) of a user–item pair (e.g., IDs), PM would output the prediction of how likely a user likes an item: $\hat{\xi}_{ui} = \tilde{P}(r_{ui} = 1|x_u, x_i; W_r)$, where W_r denotes the parameters in PM. Here, we use the notation "$\tilde{P}$" instead of "P" as it is the predicted probability rather than the ground truth.
- EM infers how likely a user is exposed to an item: $\hat{\theta}_{ui} = \tilde{P}(o_{ui} = 1|x_u, x_i; W_o)$, where W_o denotes the parameters in EM. At the end of this section, we will detail how to implement EM.
- RM targets at reasoning the causes of implicit feedback, which can be classified into four cases. Formally, given the collected data (y_{ui}) and the prior predictions from PM and EM $(\hat{\xi}_{ui}, \hat{\theta}_{ui})$, RM deduces the posterior distribution of r_{ui}, o_{ui}—i.e., the distribution over which cases the user–item pair belonging to. RM outputs a four-dimensional vector $\Phi_{ui} = [\varphi_{ui}^{(11)}, \varphi_{ui}^{(10)}, \varphi_{ui}^{(01)}, \varphi_{ui}^{(00)}]$ corresponding to the probabilities of four cases: $\varphi_{ui}^{(ab)} = \tilde{P}(r_{ui} = a, o_{ui} = b|y_{ui})$.

3.2 Model Training

Given the Architecture of ReCRec, now the question lies on how to optimize the model. Considering that the optimization of the PM and EM depends on o_{ui} and r_{ui}, which are unobserved latent variables, direct optimization is infeasible. Consequently, the E-M algorithm [14] has been employed to optimize the three modules iteratively and alternately.

E-step. Fixing the modules PM and EM, updating RM based on the following Bayes formula:

$$P(r_{ui}, o_{ui}|y_{ui}) \propto P(y_{ui}|r_{ui}, o_{ui})P(r_{ui})P(o_{ui}). \tag{7}$$

That is, we update Φ with:

$$\begin{aligned}\varphi_{ui}^{(11)} &\leftarrow y_{ui}\\ \varphi_{ui}^{(10)} &\leftarrow \frac{\hat{\xi}_{ui}(1-\hat{\theta}_{ui})+\alpha}{1-\hat{\xi}_{ui}\hat{\theta}_{ui}+3\alpha}(1-y_{ui})\\ \varphi_{ui}^{(01)} &\leftarrow \frac{(1-\hat{\xi}_{ui})\hat{\theta}_{ui}+\alpha}{1-\hat{\xi}_{ui}\hat{\theta}_{ui}+3\alpha}(1-y_{ui})\\ \varphi_{ui}^{(00)} &\leftarrow \frac{(1-\hat{\xi}_{ui})(1-\hat{\theta}_{ui})+\alpha}{1-\hat{\xi}_{ui}\hat{\theta}_{ui}+3\alpha}(1-y_{ui}).\end{aligned} \tag{8}$$

Here, a parameter α is introduced to circumvent numerical issues and promote stability during training. Conceptually, α can also be interpreted as the Dirichlet prior on Φ. Intriguingly, Equation (8) aligns with our intuition. When $y_{ui}=1$, it can be inferred that the user is both aware of and attracted to the item, and thus we have $\varphi_{ui}^{(11)}=1$; Conversely, when $y_{ui}=0$, the probability mass of Φ can be allocated among the other three cases, with the probability proportional to the model predictions on user preference and exposure (i.e., $\varphi^{(ab)} \propto \hat{P}(r_{ui}=a|W_r)\hat{P}(o_{ui}=b|W_r)$).

M-step. Updating PM and EM based on φ inferred by RM. We refer to the standard E-M algorithm [14] and optimize the following expected likelihood:

$$\max_{\hat{\xi},\hat{\theta}} \frac{1}{|\mathcal{D}|}\sum_{u,i\in\mathcal{D}} \mathbb{E}_{\tilde{P}(o_{ui},r_{ui}|y_{ui})}\left[P(y_{ui}|o_{ui},r_{ui})\tilde{P}(o_{ui},r_{ui}|x_u,x_i)\right]. \tag{9}$$

With re-organization, we can get the loss functions of PM and EM, respectively

$$L_{\mathrm{PM}}(\hat{\xi}) = \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}\left((\varphi_{ui}^{(10)}+\varphi_{ui}^{(11)})\delta_{ui}^{(1)} + (\varphi_{ui}^{(00)}+\varphi_{ui}^{(01)})\delta_{ui}^{(0)}\right) \tag{10}$$

$$L_{\mathrm{EM}}(\hat{\theta}) = \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}\left((\varphi_{ui}^{(01)}+\varphi_{ui}^{(11)})\delta^{(1)}(\hat{\theta}_{ui}) + (\varphi_{ui}^{(00)}+\varphi_{ui}^{(10)})\delta^{(0)}(\hat{\theta}_{ui})\right). \tag{11}$$

L_{PM} can be easily understood. Note that the Cases#1,#3 signify user positive preference, while Cases#2,#4 reflect negative. It would be natural to weight the positive loss $\delta_{ui}^{(1)}$ (or negative loss $\delta_{ui}^{(0)}$) with the probability that the instance belongs to Cases#1,#3 (or Cases#2,#4). Similar conclusion can be applied for L_{EM}, which targets at learning user exposure. It is noteworthy that, similar to ExpoMF, our RecRec method also incorporates an exposure model. However, unlike ExpoMF, our approach utilizes the exposure model to deduce the specific case to which an instance belongs and takes all four cases into account to learn user preference effectively. As can be seen, different from the existing methods, our ReCRec can extract both positive and negative preference signals from the unclicked data. Each unclicked instance can be classified into three cases and accordingly employ the positive or negative loss for the instance. But existing methods just employ the negative loss for the unclicked data. This key difference makes ReCRec enjoy better properties and higher recommendation accuracy.

3.3 Theoretical Analyses

In this section, we conduct theoretical analyses to answer the following important questions:

(1) *Does our model theoretically achieve unbiased estimation of the ideal loss?*
(2) *Does our model achieve lower variance comparing with the IPW?*

3.3.1 Theoretical Unbiasedness of ReCRec. To provide the answer of the first question, we first prove the unbiasedness holds when Φ_{ui} takes the proper values, and then prove the developed training procedure would push the Φ_{ui} toward the ideal. In fact, we have the following lemmas:

Lemma 2 (Unbiasedness). *When Φ_{ui} is equal to the factual posterior, i.e., $\varphi_{ui}^{(ab)} = P(r_{ui} = a, o_{ui} = b|y_{ui})$, L_{PM} is an unbiased estimation of the ideal Loss L_{Ideal}.*

Proof. For better description, we mark the loss for each instance as $l_{ui} = (\varphi_{ui}^{(10)} + \varphi_{ui}^{(11)})\delta_{ui}^{(1)} + (\varphi_{ui}^{(00)} + \varphi_{ui}^{(01)})\delta_{ui}^{(0)}$. When Φ_{ui} takes the proper values, we have

$$\begin{aligned}\mathbb{E}_{P(y_{ui})}[l_{ui}] &= \mathbb{E}_{P(y_{ui})}[(\varphi_{ui}^{(10)} + \varphi_{ui}^{(11)})\delta_{ui}^{(1)} + (\varphi_{ui}^{(00)} + \varphi_{ui}^{(01)})\delta_{ui}^{(0)}] \\ &= \mathbb{E}_{P(y_{ui})}[P(r_{ui} = 1|y_{ui})\delta_{ui}^{(1)} + P(r_{ui} = 0|y_{ui})\delta_{ui}^{(0)}] \\ &= \xi_{ui}\delta_{ui}^{(1)} + (1 - \xi_{ui})\delta_{ui}^{(0)}.\end{aligned}$$

Thus, we have $\mathbb{E}_{P(y)}[L_{\text{PM}}] = L_{\text{Ideal}}$ held. The lemma gets proof. □

Lemma 3 (Adaptivity). *With the training process proceeding, Φ_{ui} would approach to the factual posterior.*

Proof. Recent literature [2] tells us that E-M algorithm is equivalence to maximizing the following evidence lower bound:

$$L_E = \frac{1}{|D|}\sum_{u,i\in D}\mathbb{E}_{P(y)}\mathbb{E}_{\tilde{P}(r,o|y)}[\log P(y|r,o)\tilde{P}(r,o) - \log\tilde{P}(r,o|y)],$$

here the subscript ui for each variable (e.g., y_{ui}, r_{ui}, o_{ui}) is omitted for brevity. Suppose $\tilde{P}(r,o|y)$ and $P(r,o|y)$ for all user-item pairs have the same support set with bounded divergence $B_1 \le \tilde{P}(r,o|y)/P(r,o|y) \le B_2$. We have the following tight upper bound of the above evidence:

$$\begin{aligned}L_E = &\frac{1}{|D|}\sum_{u,i\in D}\Big(\mathbb{E}_{P(y)}\mathbb{E}_{P(r,o|y)}\Big[\frac{\tilde{P}(r,o|y)}{P(r,o|y)}\log P(y|r,o)\tilde{P}(r,o) - \frac{\tilde{P}(r,o|y)}{P(r,o|y)}\log P(y,r,o)\Big] \\ &+ \mathbb{E}_{P(y)}\mathbb{E}_{\tilde{P}(r,o|y)}\Big[\log P(r,o|y) + \log P(y) - \log\tilde{P}(r,o|y)\Big]\Big) \\ \le &\frac{1}{|D|}\sum_{u,i\in D}\Big(-B_1\cdot D_{KL}\big(P(y,r,o)||P(y|r,o)\tilde{P}(r,o)\big) - D_{KL}\big(\tilde{P}(r,o|y)||P(r,o|y)\big)\Big) + C \\ \le &\, C,\end{aligned}$$

where $C = \mathbb{E}_{P(y)}[\log P(y)]$ is a constant irrelevant with the optimization. The first inequality holds as $\tilde{P}(r,o|y)/P(r,o|y)$ is lower bounded with B_1, while the second inequality holds as the KL-divergence is lower bounded with 0. As can be seen, L_E achieves optimum if and only if the estimated $\tilde{P}(r,o)$ and $\tilde{P}(r,o|y)$ equal to the factual $P(r,o)$ and $P(r,o|y)$, respectively. As the training procedure of our ReCRec would improve L_E toward the optimum, it would naturally push $\tilde{P}(r,o|y)$ toward $P(r,o|y)$. Admittedly, the optimum may not be reached in practical due to the stochastic optimization or imperfect preference/exposure models. But, our empirical studies show that the approximate $\tilde{P}(r,o|y)$ is sufficient to yield good performance.

□

Remark 1. These lemmas reveal the unbiasedness and adaptivity of ReCRec—ReCRec provides an unbiased estimator of the ideal loss with the ideal Φ_{ui}, while the developed learning procedure of ReCRec would push the learned Φ_{ui} toward the ideal. ReCRec enjoys better theoretical properties over existing methods: (1) Compared with WMF and ExpoMF, ReCRec achieves unbiasedness while

WMF and ExpoMF fall short. (2) Compared with IPW, ReCRec has an adaptive learning algorithm to find the optimal configure while IPW requires extra design of the propensity.

3.3.2 Lower Variance of ReCRec. To provide the answer to the second question, we prove that the variance of our method has a finite upper bound. In fact, we have:

Lemma 4. *The variance of our estimator is*

$$\mathbb{V}\left[L_{\mathrm{PM}}\right] = \frac{1}{|\mathcal{D}|^2} \sum_{(u,i)\in\mathcal{D}} \mathbb{V}\left[\varphi_{ui}^{(10)} + \varphi_{ui}^{(11)}\right]\left(\delta_{ui}^{(1)} - \delta_{ui}^{(0)}\right)^2, \tag{12}$$

which is bounded with

$$\mathbb{V}\left[L_{\mathrm{PM}}\right] \le \frac{1}{4} \cdot \frac{1}{|\mathcal{D}|^2} \sum_{(u,i)\in\mathcal{D}} \left(\delta_{ui}^{(1)} - \delta_{ui}^{(0)}\right)^2. \tag{13}$$

Proof. For convenient, let $h_{ui} = \varphi_{ui}^{(10)} + \varphi_{ui}^{(11)}$. We have

$$\begin{aligned}
\mathbb{E}\left[(l_{ui})^2\right] &= \mathbb{E}\left[(h_{ui})^2\right]\left(\delta_{ui}^{(1)}\right)^2 + \mathbb{E}\left[(1-h_{ui})^2\right]\left(\delta_{ui}^{(0)}\right)^2 + 2\left(\mathbb{E}\left[h_{ui}\right] - \mathbb{E}\left[(h_{ui})^2\right]\right)\delta_{ui}^{(1)}\delta_{ui}^{(0)} \\
\mathbb{E}^2\left[l_{ui}\right] &= \mathbb{E}^2[h_{ui}]\left(\delta_{ui}^{(1)}\right)^2 + 2\mathbb{E}[h_{ui}]\left(1-\mathbb{E}[h_{ui}]\right)\delta_{ui}^{(1)}\delta_{ui}^{(0)} + \left(1-\mathbb{E}[h_{ui}]\right)^2\left(\delta_{ui}^{(0)}\right)^2.
\end{aligned} \tag{14}$$

So, we have

$$\begin{aligned}
\mathbb{V}[l_{ui}] &= \mathbb{E}\left[(l_{ui})^2\right] - \mathbb{E}^2[l_{ui}] \\
&= \left(\mathbb{E}\left[(h_{ui})^2\right] - \mathbb{E}^2\left[h_{ui}\right]\right)\cdot\left(\delta_{ui}^{(1)} - \delta_{ui}^{(0)}\right)^2 \\
&= \mathbb{V}\left[h_{ui}\right]\cdot\left(\delta_{ui}^{(1)} - \delta_{ui}^{(0)}\right)^2.
\end{aligned} \tag{15}$$

Thus, the variance of L_{PM} can be written as follows:

$$\mathbb{V}\left[L_{\mathrm{PM}}\right] = \frac{1}{|\mathcal{D}|^2}\sum_{(u,i)\in\mathcal{D}}\mathbb{V}\left[l_{ui}\right] = \frac{1}{|\mathcal{D}|^2}\sum_{(u,i)\in\mathcal{D}}\mathbb{V}\left[\varphi_{ui}^{(10)} + \varphi_{ui}^{(11)}\right]\left(\delta_{ui}^{(1)} - \delta_{ui}^{(0)}\right)^2. \tag{16}$$

Note that h_{ui} takes the value on $[0,1]$, $\mathbb{V}\left[h_{ui}\right] \le \frac{1}{4}$, we have

$$\mathbb{V}\left[L_{\mathrm{PM}}\right] = \frac{1}{|\mathcal{D}|^2}\sum_{(u,i)\in\mathcal{D}}\mathbb{V}\left[l_{ui}\right] \le \frac{1}{4\cdot|\mathcal{D}|^2}\sum_{(u,i)\in\mathcal{D}}\left(\delta_{ui}^{(1)} - \delta_{ui}^{(0)}\right)^2.$$

□

Remark 2. This lemma demonstrates our ReCRec usually has lower variance than IPW. By comparing Equation (6) and Equation (12), the variance in ReCRec is bounded with a small value while it may be exploded in IPW especially for the instances with low exposure probability.

3.4 Implementations of EM

The architecture of EM is important for ReCRec. EM should be different from PM with encoding human prior knowledge of exposure into the module. Here, we provide two scenes:

(1) *Item-based exposure model (ReCRec-I)* assumes user exposure only depends on item own characteristics. It simply introduces an item-dependent learnable parameter μ_i and model EM with $\hat{\theta}_{ui} = \sigma(b_i)$, where $\sigma(.)$ denotes the Sigmoid function.

(2) *Factorization-based exposure model (ReCRec-F)* considers the exposure depends on both user and item. Given that different users usually experience different recommendation policy from the system, the exposure probability should be naturally personalized. That is, the opportunity

for different users to be exposed to an item would vary. A scalar item-based term is insufficient for modelling exposure. Thus, here we model the exposure with user/item-based latent factors: $\hat{\theta}_{ui} = \sigma\left(\mathbf{e}_u^\top \mathbf{e}_i^\top + b_u + b_i\right)$, where $\mathbf{e}_u, \mathbf{e}_i, b_u, b_i$ are learnable parameters *w.r.t.* user or item. Specifically, when training EM, a regularizer is introduced to constrain $\hat{\theta}_{ui}$ to be close to the item popularity, *i.e.*, $||\hat{\theta}_{ui} - s_i||^2$. Here, s_i denotes the normalized item popularity and can be calculated by $s_i = (g_i/g_{max})^{0.5}$, where g_i denotes the popularity of the item i and g_{max} denotes the largest popularity among all items. The introduced regularizer is based on our intuition that items with greater popularity are more likely to be exposed to users.

4 Semi-Synthetic Experiment

In this section, we conduct experiments on a semi-synthetic dataset. Unlike real-world datasets, the semi-synthetic dataset contains the ground truth of user preference and exposure, allowing us to explore the following **research questions (RQs)**:

— *RQ1*: How does ReCRec learn user preference and exposure compared with existing methods?
— *RQ2*: How is the variance of ReCRec compared with IPW?

4.1 Experiment Setup

Dataset. We closely refer to [53] in generating the semi-synthetic dataset for fair comparison. We use the MovieLens(ML)-100K dataset and generate ground-truth preference and exposure via **matrix factorization (MF)**. A specific parameter p is introduced to control the skewness of the distribution of the exposure. A larger p suggests larger exposure bias. The details are as follows:

(1) Using MF [33] to get an approximation of the true ratings $\hat{a}_{ui}$.
(2) Using logistic MF model [31] to get an approximation of the true observations $\hat{b}_{ui}$.
(3) Generating the ground-truth parameters of preference and exposure as follows:

$$\begin{aligned}\xi_{ui} &= \sigma(\hat{a}_{ui} - \epsilon)\\ \theta_{ui} &= \frac{(\hat{b}_{ui})^p}{\sum\limits_{(u,i)\in\mathcal{D}} (\hat{b}_{ui})^p} \cdot C,\end{aligned} \tag{17}$$

where $\sigma(.)$ is the sigmoid function, ϵ controls the overall level of user preference, and p controls the skewness of exposure. The larger ϵ would make the dataset have less positive feedback, while the larger p suggests the larger exposure bias introduced in the data, where the difference in b_{ui} has been amplified. In the experiment, we refer to recent work [53] and set $\epsilon = 5$, and test the model performance with varying $p = 0.5, 1.0, 1.5$. Note that too large p will cause most of the exposure parameters θ_{ui} to be too small, resulting in highly sparse dataset. Here, we re-scale θ_{ui} to make their sum as a constant C. Here, we simply set C as the benchmark when p is set to 0.5, *i.e.*, $C = \sum\limits_{(u,i)\in\mathcal{D}} (\hat{b}_{ui})^{0.5}$.

(4) Generating the click variables via Bernoulli sampling:

$$\begin{aligned}o_{ui} &\sim Bern(\theta_{ui})\\ r_{ui} &\sim Bern(\xi_{ui})\\ y_{ui} &= o_{ui} \cdot r_{ui}.\end{aligned} \tag{18}$$

Given the synthetic data on all user–item pairs, we randomly sampled all clicked data, with 90% serving as the training set and 10% as the validation set. The remaining unclicked data was used as the test set, and the preference variable r_{ui} from the unclicked data served as the ground truth during testing.

We chose all the clicked data and randomly sampled 90% and 10% for model training and validation, and the remaining data were used as the test set, with the preference variables r_{ui} from the test data serving as the ground truth during testing.

Baseline. We compare ReCRec with the following baselines:

— WMF [28]: the classic WMF that down-weighs the contribution of unclicked data. To make the baseline stronger, we experimented with several weighting strategies, which included uniform [28], item-popularity-based [24], and user-activity-based strategies [46]. We report the best performance.
— ExpoMF [38]: the representative debiasing method with leveraging an exposure-based generative model.
— Rel-MF [53]: the representative debiasing method based on propensity. Rel-MF leverages item popularity to specify the propensity.
— CJMF [76]: the state-of-the-art propensity-based debiasing method that leverages a symmetric learning strategy to jointly learn the propensity and the user preference.
— BISER [35]: the state-of-the-art propensity-based method that specifies the propensity based on the prediction from the preference model.

For fair comparison, all of these methods are based on the uniform MF backbone model, with the exception of BISER. BISER utilizes Auto-encoder, as suggested by original paper, which is stronger than MF. We also try BISER with MF but got inferior performance.

Evaluation Metrics. We utilize the following metrics to evaluate our method:

— *Logloss* evaluates the accuracy of the predictions on preference or exposure with:

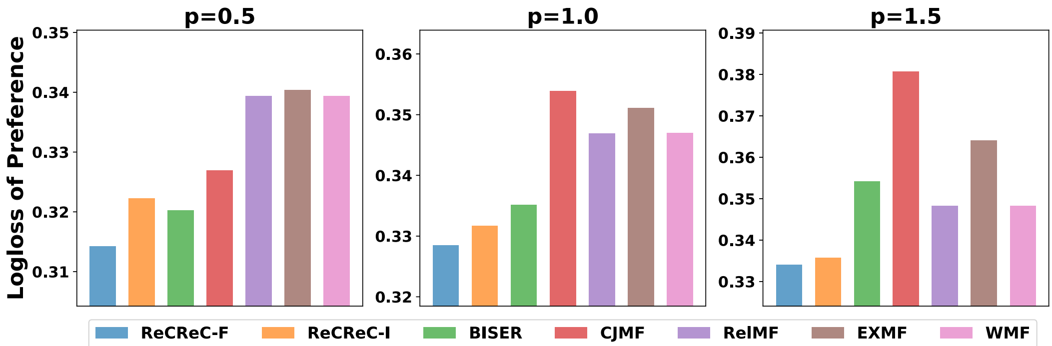
$$Logloss(\xi) = \frac{1}{|\mathcal{D}_t|} \sum_{u,i \in \mathcal{D}_t} (-\xi_{ui} \log(\hat{\xi}_{ui}) - (1 - \xi_{ui}) \log(1 - \hat{\xi}_{ui}))$$
$$Logloss(\theta) = \frac{1}{|\mathcal{D}_t|} \sum_{u,i \in \mathcal{D}_t} (-\theta_{ui} \log(\hat{\theta}_{ui}) - (1 - \theta_{ui}) \log(1 - \hat{\theta}_{ui})), \tag{19}$$

where $\mathcal{D}_t$ denotes the set of user–item pairs in the test data.
— *NDCG@K* measures the ranking quality of recommendation through discounted importance based on the position

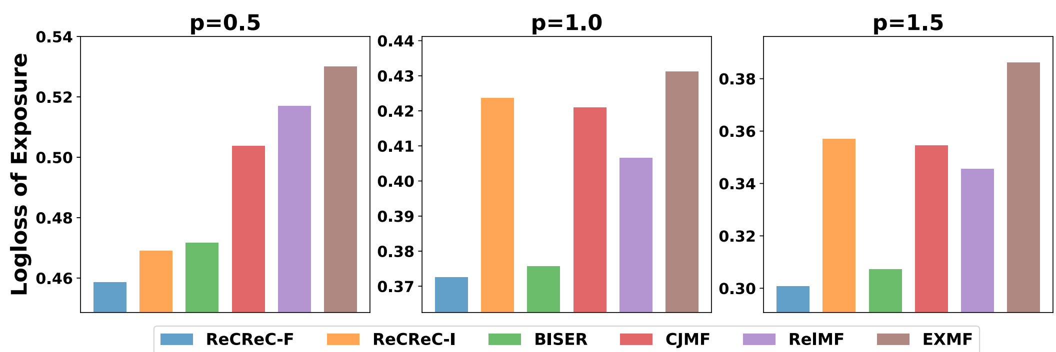
$$DCG_u@K = \sum_{i \in \mathcal{S}(u)} \frac{\mathrm{I}[\hat{Z}_{ui} \le K]}{\log\left(\hat{Z}_{ui} + 1\right)}$$
$$NDCG@K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{DCG_u@K}{IDCG_u@K}, \tag{20}$$

where I[] denotes an indicator function, $\mathcal{S}(u)$ denotes the positive item set in the test data for a user u, and $\hat{Z}_{ui}$ denotes the ranking position of the item i when ranking the items on the test data for the user u. $IDCG_u@K$ is the $DCG_u@K$ value of the ideal ranking with the optimal ranking for the user u.
— *Empirical variance* evaluates the variance of the estimator with

$$\hat{\mathbb{V}}[L(\hat{\xi})] = \frac{1}{|\mathcal{D}|} \sum_{u,i \in \mathcal{D}} (P(y_{ui} = 0)(L(\hat{\xi}|y_{ui} = 0) - \bar{L}_{ui})^2 + P(y_{ui} = 1)(L(\hat{\xi}|y_{ui} = 1) - \bar{L}_{ui})^2), \tag{21}$$



(a) Logloss on preference estimation.



(b) Logloss on exposure estimation.

Fig. 3. Comparison of debiasing methods with varying values of p, where p controls the severity of exposure bias: (a) Logloss on test data for measuring the accuracy of capturing user preference; (b) Logloss for measuring the accuracy of capturing user exposure. The smaller Logloss signifies better prediction accuracy.

where $\bar{L}_{ui}$ denotes the mean of the loss calculated by

$$\bar{L}_{ui} = P(y_{ui} = 0)L(\hat{\xi}|y_{ui} = 0) + P(y_{ui} = 1)L(\hat{\xi}|y_{ui} = 1), \tag{22}$$

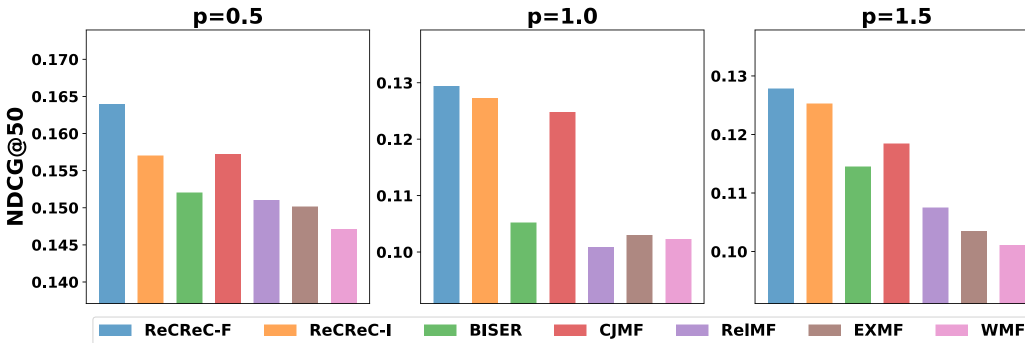
where $P(y_{ui} = 1) = \theta_{ui}\xi_{ui}$ and $P(y_{ui} = 0) = 1 - \theta_{ui}\xi_{ui}$.

Hyperparameter Settings. We implement our model in Tensorflow. Adam is adopted as our optimization method. Grid search is utilized to find best hyperparameters. The search range of the learning rate for all experiments is $\{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\}$ and the range of the L2 regularization coefficient is $\{10^{-6}, 10^{-5}, ..., 0.1\}$. The batch size is set to 2^{12}. The embedding size is fixed to 20 for all methods as in recent work [53]. For the compared baselines, we closely follow their settings reported in the relevant papers or directly utilize their codes if they are available. We also have finely tuned their hyperparameters to ensure optimal performance.

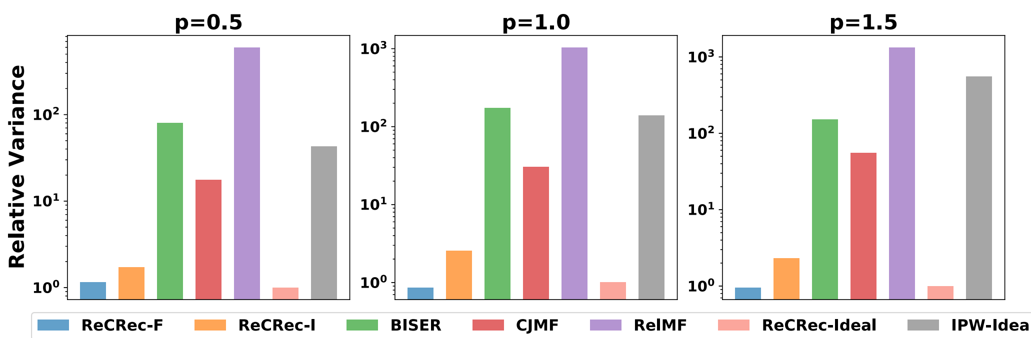
4.2 Performance Comparison (RQ1)

The results are presented in Figures 3, 4, and 5. We make the following observations:

— Compared with existing methods, ReCRec achieves much better accuracy in predicting user preference and exposure (cf. Figure 3(a) and (b)). More impressively, as the level of exposure bias increases (i.e., p becomes larger), the margin of the improvement becomes larger. The



(a) Ranking performance comparison.



(b) Relative variance comparison.

Fig. 4. Comparison of debiasing methods with varying values of p, where p controls the severity of exposure bias: (a) ranking performance (NDCG@50) comparison; (b) variance comparison between ReCRec and IPW. As the variance may have a wide range of magnitude, here we present the variance relative to ReCRec-ideal for better illustration. ReCRec-ideal (or IPW-ideal) denotes the method with the ideal distribution Φ_{ui} (or propensity ρ).

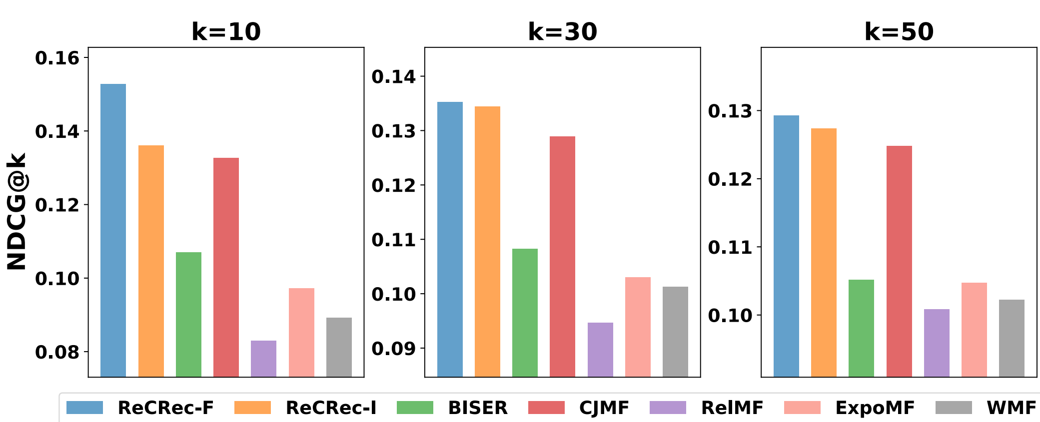


Fig. 5. Comparison of debiasing methods with varying values of K in $NDCG@K$ at $p = 1$.

advancement can be attributed to the superior theoretical attributes of ReCRec, which exhibits the potential to achieve unbiasedness while maintaining low variance. The accuracy of prediction also brings better recommendation performance. It can be seen that ReCRec also has better NDCG@50 than others (cf. Figure 4(a)). According to Figure 5, for different values of K, the $NDCG@K$ of RecRec surpasses all comparison methods, further validating the effectiveness of our approach.
—ReCRec-I performs significantly worse than ReCRec-F on predicting user exposure. This outcome reveals the limitation of an item-based exposure model in capturing complex patterns. The process of exposure should be personalized, as the probability for different users to encounter a particular item varies. Consequently, ReCRec-F that leverages personalized exposure model exhibits better accuracy than ReCRec-I in capturing user exposure, leading to better debiasing recommendation performance.

4.3 Variance Comparison (RQ2)

Figure 4(b) illustrates the variance of IPW-based methods and our ReCRec. We make the following obervations:

—The variance of ReCRec is indeed much smaller than IPW-based methods—usually reducing by more than 10 times. This result clearly demonstrates the advantage of ReCRec—it could avoid the high variance weakness of IPW and thus yield more stable and better recommendation performance (cf. Figure 4(b)).
—To our surprise, we can find that ReCRec (or IPW) with estimated parameters Φ (or propensity) might have better variance than ideal. It can be seen from ReCRec-F has lower variance than ReCRec-ideal, and CJMF has lower variance than IPW-ideal. This interesting phenomenon may be attributed to the constraint capacity of the exposure model. Both ReCRec-F and CJMF introduce strong constraints on the exposure model, i.e., aligning with item popularity. This consequently curtails the flexibility of the exposure model and correspondingly limits the variance of Φ (or propensity ρ). The variance of the estimator is reduced but may incur more bias.

5 Real-World Experiment

In this section, we conduct extensive experiments on real-world datasets to validate the superiority of our ReCRec. We aim to answer the following major research questions:

—*RQ3*: Does ReCRec outperform existing debiasing methods in real-world datasets?
—*RQ4*: How does ReCRec learn user exposure in real-world datasets?
—*RQ5*: How does each module of ReCRec impact performance?
—*RQ6*: How do hyperparameters (e.g., embedding dimension, regularizer coefficient) affect model performance?

5.1 Experimental Setup

Datasets. We closely refer to [35] and use three public real-world datasets (Yahoo!R3, Coat, and KuaiRand-Pure) for experiments. All three datasets contain a set of biased rating data collecting the normal interactions of users in the platform, and a set of unbiased rating data from stochastic experiment. Following [35], we utilize biased data for model training and unbiased data for model validation (20%) and test (80%). Also, the rating data are translated into implicit feedback, i.e., the ratings larger than three is regarded as positive otherwise as negative. The detailed information on the datasets is shown in Table 2.

Table 2. Dataset Statistics

Statistics	Yahoo!R3	Coat	KuaiRand-Pure
#Users	15,400	290	27,285
#Items	1,000	300	7,583
#Biased ratings	311,704	6,960	1436,609
#Unbiased ratings	54,000	4,640	1186,059

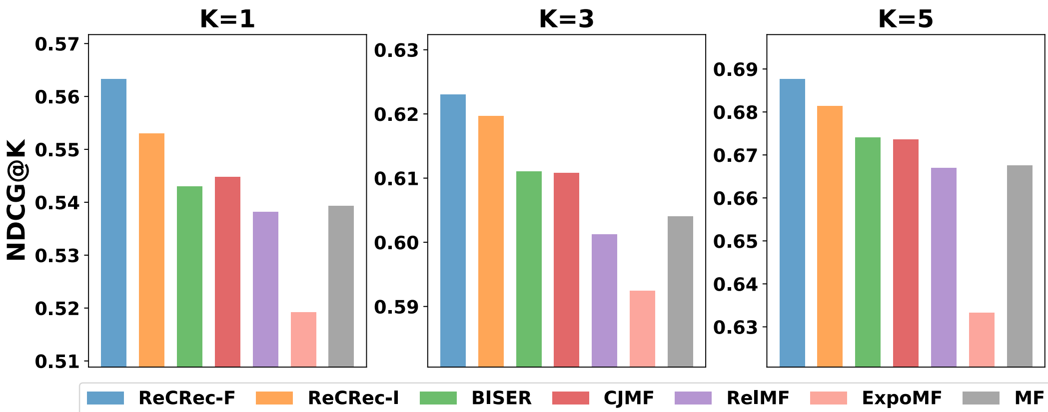


Fig. 6. Comparison of *NDCG@K* metrics at different values of K.

Evaluation Metrics. Besides *NDCG@K*, the following additional ranking metrics are employed for evaluating recommendation performance as recent work [53]:

— *Recall@K* quantifies the proportion of positive items found in the top-K recommended items

$$Recall@K = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{S}(u)}\frac{r_{ui}\mathrm{I}[\hat{Z}_{ui}\leq K]}{|S(u)|}, \tag{23}$$

where I[] denotes an indicator function, $\mathcal{S}(u)$ denotes the positive item set in the test data for a user u, and $\hat{Z}_{ui}$ denotes the ranking position of the item i when ranking the items on the test data for the user u.

— *MAP@K* denotes mean average precision calculated by

$$MAP@K = \frac{1}{K|\mathcal{U}|}\sum_{k=1}^{K}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{S}(u)}\frac{r_{ui}\mathrm{I}[\hat{Z}_{ui}\leq k]}{k}. \tag{24}$$

Considering a few positive instances in the test data, here we refer to recent work [53] and set K as 5. More details about the impact of K can refer to Figure 6.

Hyperparameter Settings. The hyperparameter settings of the real-world experiment are similar with the ones of semi-synthetic experiment, except that the embedding dimension is fixed to 200 for all methods in the real-world experiment. We find a larger embedding dimension in real-world datasets can significantly enhance recommendation performance. Setting the embedding dimension at 20, as performed in the semi-synthetic experiment, appears to be inadequate. More details about the impact of dimensions can be found in Figure 9.

Table 3. Performance Comparison between Our ReCRec and Baselines on Three Real-World Datasets

	Yahoo!R3			Coat			KuaiRand		
Model	NDCG@5	MAP@5	Recall@5	NDCG@5	MAP@5	Recall@5	NDCG@5	MAP@5	Recall@5
MF	0.6685	0.2234	0.7972	0.5529	0.2645	0.5565	0.3783	0.2026	0.3180
ExpoMF	0.6233	0.2084	0.7508	0.5247	0.2533	0.5419	0.3591	0.1964	0.3102
Rel-MF	0.6669	0.2230	0.7953	0.5557	0.2709	0.5463	0.3800	0.2055	0.3199
CJMF	0.6727	0.2257	0.8005	0.5291	0.2632	0.5319	0.3918	0.2144	0.3230
BISER[a]	0.6741	0.2260	0.8050	0.5577	0.2748	0.5422	0.3925	0.2166	0.3232
ReCRec-I	0.6818	0.2289	0.8114	0.5865	**0.2856**	**0.5800**	0.3946	0.2170	0.3269
ReCRec-F	**0.6872**	**0.2327**	**0.8124**	**0.5946**	0.2845	0.5787	**0.3979**	**0.2184**	**0.3285**
Impv-b (%)	1.9%[b]	2.1%[b]	0.9%[b]	6.6%[b]	3.9%[b]	7.0%[b]	1.37%	0.81%	1.63%

The best results are marked in boldface. The row 'Impv-b' indicates the relative performance gain of our ReCRec-F compared to BISER. [a]Indicates the best baseline model, i.e., BISER. [b]Indicates that the improvement is significant with t-test at $p < 0.05$.

5.2 Performance Comparison (RQ3)

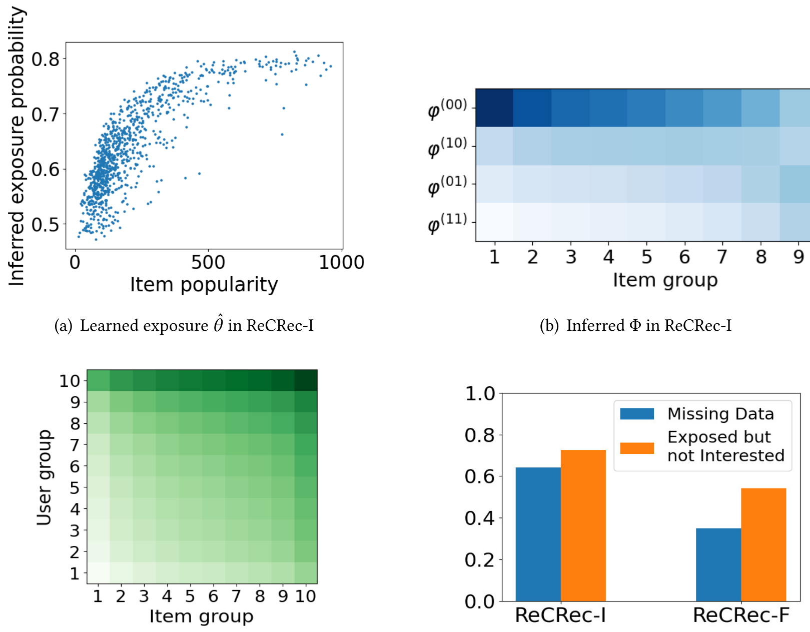
Table 3 and Figures 6 and 9 show the overall performance of our ReCRec compared with other debiasing methods. We make the following observations:

— Our ReCRec-F and ReCRec-I consistently outperform other baselines on all three datasets in terms of all metrics. Especially in the dataset Coat, the improvements are encouraging. ReCRec-F achieves 6.6%, 3.9%, and 7.0% performance gain over the best baseline in terms of NDCG@5, MAP@5, and Recall@5, respectively. This result clearly demonstrate the superiority of the proposed ReCRec. Its theoretical advantages indeed bring better empirical better recommendation performance on real-world datasets.
— In terms of two different EM, we observe ReCRec-F is relatively better ReCRec-I. This result confirms that modeling personalized exposure is necessary and potentially enhance recommendation performance.
— Figure 6 presents a comparison of $NDCG@K$ for all methods at different values of K. It can be observed that both ReCRec-I and ReCRec-F surpass all other comparison methods across different K values, further demonstrating the superiority of our ReCRec method.

5.3 Exploratory Analysis (RQ4)

We now explore the distributions of the inferred exposure of ReCRec on the Yahoo!R3 dataset. We conduct the following empirical analyses:

— Figure 7(a) illustrates the learned item exposure probability ($\hat{\theta}_{ui}$) with item popularity. We observe that the item with a higher popularity has a larger θ_{ui}. ReCRec-I could adaptively capture the positive relation between the popularity and exposure without requiring manual specifying.
— Figure 7(b) shows the inferred distribution over four cases (Φ) with the item popularity. For better presentation, items are segmented into ten groups based on their popularity referring to [74]. The lager group ID suggests larger popularity. The average values are then calculated within each group. We make the following observations: (1) The average $\varphi^{(01)}$ and $\varphi^{(11)}$ increases with the larger popularity. This can be attributed to the fact that higher popularity typically correlates with increased exposure opportunities. (2) The value of $\varphi^{(10)}$ exhibits an interesting pattern. It initially rises and then reduces as item popularity increases. This unique



(a) Learned exposure $\hat{\theta}$ in ReCRec-I

(b) Inferred Φ in ReCRec-I

(c) Learned exposure $\hat{\theta}$ in ReCRec-F

(d) Learned exposure $\hat{\theta}$ on exposed/missing data

Fig. 7. Exploring the distributions of the inferred exposure on Yahoo!R3.

phenomenon can be attributed to two main factors: (a) Popular items generally possess superior quality and are therefore more likely to appeal to users, leading to an initial increase in $\varphi^{(10)}$; (b) However, once the popularity surpasses a certain threshold, items become exceedingly likely to be exposed. This impact overwhelms the former effect, culminating in a decrease in $\varphi^{(10)}$. These results validate the adapativity of ReCRec. It indeed captures some useful data patterns.

— Does ReCRec-F capture personalized exposure? To give the answer, we refer to [74] and divide users and items into 10 groups according to item popularity and user activity. The average exposure for each group is shown in Figure 7(c). The darker the color indicates the larger value. We observe ReCRec could capture the personalized exposure, i.e., the more active users are prone to be exposed to more items.

— Note that Yahoo!R3 dataset contains a small dataset where a user is exposed but dislikes an item (i.e., the item with low rating value). Thus, we explore model prediction of exposure on this small exposed dataset and other missing data. The average predicted exposure probability is shown in Figure 7(d). We observe (1) the score on exposed data is larger than missing data, suggesting ReCRec indeed captures exposure to a certain degree; (2) the discrepancy in scores between exposed and missing data for ReCRec-F is larger than that for ReCRec-I, implying that ReCRec-F is comparatively more accurate than ReCRec-I in capturing user exposure.

5.4 Ablation Study (RQ5)

We now examine the impact of each module within ReCRec: (1) initially, we exclude the RM, precluding the use of knowledge from the EM for learning user preferences, which would reduce the model to a basic WMF; (2) we then assess the significance of different cases in learning user

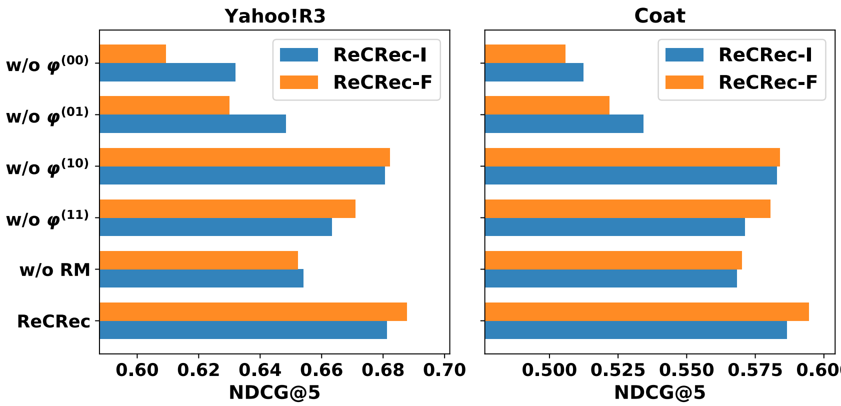
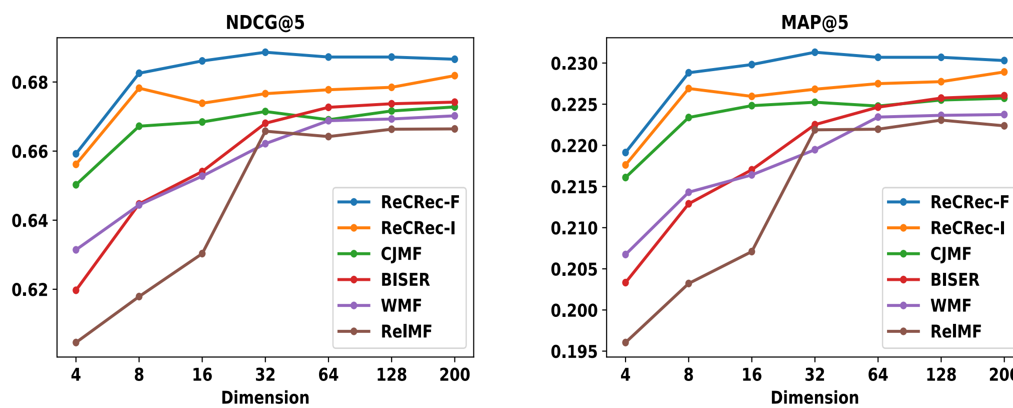


Fig. 8. The impact of each component of RM on the performance of ReCRec on Yahoo!R3 and Coat.



(a) NDCG@5 with varying dimensions

(b) MAP@5 with varying dimensions

Fig. 9. Recommendation performance of the compared debiasing methods with different embedding dimensions on the Yahoo!R3 dataset.

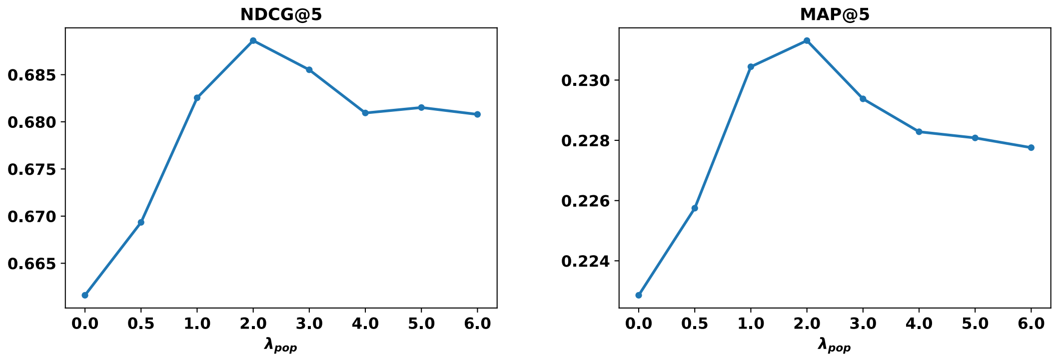
preference. Toward this end, we individually omit each case (i.e., setting $\varphi^{(11)}$, $\varphi^{(10)}$, $\varphi^{(01)}$ or $\varphi^{(00)}$ as zero in the objective L_{PM}) in learning user preference.

The results are presented in Figure 8. As can be seen, the removal of either RM or any φ results in diminished performance. This finding underscores the criticality of considering all four cases when learning user preferences from implicit feedback.

5.5 Hyperparameter Studies (RQ6)

5.5.1 Impact of Embedding Dimension. The diagram depicted in Figure 9 provides a comparative analysis of the performance of various methods with varying embedding dimensions. For all methods, we adjust the embedding size over a range of values 4, 8, 16, 64, 128, 200, using the Yahoo!R3 dataset.

From the results, it is evident that both ReCReC-I and ReCRec-F consistently surpass the performance of other models across all examined dimensions. Another phenomenon is observed wherein the performance of all methods improves and eventually plateaus as the dimensionality escalates. It



(a) NDCG@5 with varying λ_{pop} (b) MAP@5 with varying λ_{pop}

Fig. 10. Impact of λ_{pop} (the coefficient of popularity regularizer) on the performance of ReCRec-F on the Yahoo!R3 dataset.

is worth noting that our RecRec achieves near-optimal results and reaches a state of stability when the dimensionality is eight. This observation suggests that our approach is capable of extracting meaningful and efficient information even though using a limited dimensional space.

5.5.2 Impact of Coefficient of Popularity Regularizer. In the ReCRec-F model, λ_{pop}, the coefficient of the popularity regularization term controls the degree of similarity between the predicted exposure and the popularity of items. We conduct experiments on the Yahoo!R3 dataset to study how the value of λ_{pop} affect recommendation performance (NDCG@5, MAP@5). The results are shown in Figure 10.

As we can be seen, as λ_{pop} becomes larger, there is an initial improvement in performance. This can be attributed to the fact that user exposure is indeed positively correlated with item popularity. Consequently, incorporating such prior knowledge can enhance the learning process of exposure. However, when λ_{pop} surpasses a certain threshold (e.g., $\lambda_{pop} = 2.0$), the performance deteriorates with further increases in λ_{pop}. This can be explained by the fact that overly stringent constraints on exposure can compromise its flexibility and personalization, leading to the sub-optimal performance. Thus, a tradeoff for λ_{pop} is observed. When λ_{pop} is set to a proper value (e.g., $\lambda_{pop} = 2.0$), the model achieves the best performance.

6 Related Work

In this section, we briefly review related work from the following three perspectives.

6.1 Recommendation Models

Recent years have witnessed the flourish publications on recommendation models. The most classic recommendation model is MF [33, 47, 57], which projected IDs of users and items into embedding vectors and then recovered the feedback via the inner product of the user and item embedding. As the inner product violates triangle inequality and lacks generalization [27], some recent work proposed to utilize more advanced distance metrics for generating interactions. For example, CML [27] utilized the Euclidean distance on the embedding space; LRML [56] introduced extra relation vectors to evaluate the relations between users and items. Besides, neural network has been leveraged to capture complex patterns of user–item interactions, spanning from Multilayer Perceptron [23], Convolutional Neural Network [22], and AutoEncoder [39, 43, 54, 55].

For enriching the learning of user representation, some researchers have incorporated users' historical behaviors into their models. The most representative methods are FISM and SVD++, which pool the embeddings of the interacted items to generate the user embedding. More recently, the sequence of item interactions has received substantial attention, sparking a burgeoning field of sequential recommendation. Various techniques have been proposed to encode the users' historical behaviors. These range from the traditional sequential probabilistic models [51] to recent advanced Recurrent Neural Networks [26], transformer architectures [11, 32], Neural differential Models [20], and LLM-based Models [13].

Given the effect of **graph neural network (GNN)** [16, 68] on capturing high-order relations, GNN has also been introduced in RS to mining high-order collaborative signals in historical interactions. NGCF [62] and GC-MC [1] extended the user-item interactions into a specific bipartite graph and then performed graph propagation to integrate the information from the (high-order) neighbors into the representation learning. HGAN [63] further introduced attention mechanisms to aggregate information from different nodes in the graph; LightGCN [21] simplified the NGCF architecture by removing unnecessary components; A number of researchers [59, 66, 72] further leveraged the contrastive learning [65] in a graph-based recommendation to enhance its robustness and generalization; Some researcher [15, 58] also studied generalization and distribution shift issue in graph-based recommender.

6.2 Bias in RS

Given that data in RS are collected through observation rather than rigorous experiments, bias frequently manifests within such systems [5]. Specifically, data bias refers to a scenario where the distribution of the collected data diverges from that utilized for model testing [4]. Therefore, blindly training a recommendation model on such skewed data could not only lead to suboptimal recommendation accuracy, but could also potentially engender unfairness and intensify the Matthew effect [17, 18, 29, 37].

Recent work [5] has classified data bias into four types including (1) selection bias [25, 36, 44, 49], which happens in explicit feedback data referring to a phenomenon that the observed data might not faithfully represent the entirety of user–item pairs; (2) exposure bias [38], which happens in implicit feedback indicating a phenomenon that the click may not purely signify user preference; (3) position bias [12, 30, 45] referring to a phenomenon that users' feedback would be impacted by the item position displayed to the user; (4) conformity bias [42, 61] referring to a phenomenon that users tend to behave similarly to the others in a group.

Distinct types of biases display unique characteristics. Recently have witnessed various debiasing strategies tailored for diverse biases [9, 40, 64, 70, 71]. As this work mainly focuses on exposure bias, here we mainly review the debiasing methods for this type of bias. For other biases, readers are encouraged to consult the extensive survey [5] on this topic.

6.3 Debiasing Exposure Bias

Exposure bias happens as users are only exposed to a part of specific items so that unclicked do not always represent negative preference [5]. Exposure bias is common and may root in many factors such as item popularity [75], user background [8] or previous recommendation policy [41]. Recent work on addressing exposure bias can be mainly categorized into three types.

WMF, a classical approach, weighed unclicked data with specific confidence weights. Recent studies have suggested varied weighting strategies. For example, Hu et al. [28] downweighted unclicked data with uniformly assigning them lower scores compared to clicked data; He et al. [24] utilized popularity-based weights, under the assumption that popular items are more likely to be exposed to users; Pan and Scholz [46] further incorporates user-activity in the weighting strategy.

Another type is ExpoMF [38], which deduced how likely a user is exposed to an item through a generative exposure model and learned a recommendation model on the exposed data. Recent studies have suggested varied architecture of the exposure model. For instance, Liang et al. [38] proposed to generate the exposure based on item popularity or text topics; Chen et al. and Wang et al. [6, 8, 60] incorporated social relations in the exposure model under the assumption that social connected users share similar exposure distribution; Chen et al. [7] further took users community into consideration.

More recently, some researchers proposed to leverage the propensity score to address exposure bias. These methods offered an impressive theoretical property—achieving an unbiased estimator of the ideal loss when the propensity is correctly specified. As the success of these methods hinges on the accurate specification of the propensity, various strategies were developed. For example, Saito et al. [53] proposed to utilize the item popularity to determine propensity; Zhu et al. [76] developed a jointly learning strategy to infer both propensity and user preference; Lee et al. [35] considered to specify the propensity using the predictions from the recommendation model.

However, we argue that all of existing methods overlook the positive signals among unclicked data, incurring bias estimation or high variance. Our ReCRec deduces the causes behind unclicked data, mining both positive and negative signals and thus yielding better performance than existing methods.

Besides the above debasing methods that tailored for exposure bias, there are some general debiasing methods that can be adopted for addressing exposure bias. These methods mainly resort to a small unbiased dataset for guiding the training of the debiasing model. For example, Bonner and Vasile [3] proposed to align the embeddings learned on the biased data and unbiased data; Liu et al. [41] proposed to leverage the knowledge distillation to extract useful unbiased knowledge from the unbiased data; Chen et al. [4] proposed to learn the propensities and imputations via meta learning from the unbiased data. While these methods hold promise, they necessitate the availability of unbiased datasets. The collection of such datasets, however, is both challenging and costly. It involves the intervention of the recommendation, e.g., using the random policy, which would significantly hurt company benefits and user experience. This drawback substantially impedes the practical application of these methods.

7 Conclusion and Future Work

In this work, we introduce a new method, ReCRec, that discerns the causes of implicit feedback for debiased recommendation. ReCRec consists of a PM for capturing user preference, an EM for inferring user exposure and a RM targeting at reasoning to which case each instance belongs. A specific learning algorithm is developed to update the modules iteratively and alternatively. Rigorous theoretical analyses affirm that ReCRec could yield an unbiased estimate of the ideal loss with a bounded variance. Additionally, comprehensive experiments on both semi-synthetic and real-world datasets validate its superiority over current state-of-the-art methods.

This work opens up several intriguing avenues for future research. Firstly, while RecRec demonstrates impressive performance with a simple exposure model, there is potential to delve further into more sophisticated EM. This could involve the use of neural networks or graph-based models, which may offer increased precision in capturing user exposure and, as a result, enhance model performance. Secondly, in practical, exposure bias is dynamic rather than static. Typically, users are continuously exposed to new recommendations. Therefore, investigating temporal exposure bias by leveraging sequential models could provide a promising direction for subsequent research.

Appendix

A Proof of Lemma 1

Proof. The proof procedure refers to [53]. We can write the expectation of the objective of ExpoMF as follows:

$$\begin{aligned}\mathbb{E}[L_{ExpoMF}(\hat{\xi})] &= \mathbb{E}[\frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}\hat{\theta}_{ui}[y_{ui}\delta_{ui}^{(1)} + (1-y_{ui})\delta_{ui}^{(0)}]] \\ &= \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}\hat{\theta}_{ui}[\mathbb{E}[y_{ui}]\delta_{ui}^{(1)} + (1-\mathbb{E}[y_{ui}])\delta_{ui}^{(0)}] \\ &= \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}\hat{\theta}_{ui}\theta_{ui}\gamma_{ui}\delta_{ui}^{(1)} + \hat{\theta}_{ui}(1-\theta_{ui}\gamma_{ui})\delta_{ui}^{(0)}].\end{aligned} \tag{25}$$

Then, we have the gap between the objective of ExpoMF with the ideal loss:

$$\begin{aligned}&\mathbb{E}[L_{ExpoMF}(\hat{\xi})] - L_{ideal}(\hat{\xi}) \\ &= \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}\hat{\theta}_{ui}\theta_{ui}\gamma_{ui}\delta_{ui}^{(1)} + \hat{\theta}_{ui}(1-\theta_{ui}\gamma_{ui})\delta_{ui}^{(0)}] - \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}[\gamma_{ui}\delta_{(ui)}^{(1)} + (1-\gamma_{ui})\delta_{ui}^{(0)}] \\ &= \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}}[\gamma_{ui}(\hat{\theta}_{ui}\theta_{ui}-1)\delta_{ui}^{(1)} + \hat{\theta}_{ui} - 1 - \gamma_{ui}(\hat{\theta}_{ui}\theta_{ui}-1)\delta_{ui}^{(0)}].\end{aligned} \tag{26}$$

As can be seen, the gap becomes zero if and only if $\theta_{ui} = \hat{\theta}_{ui} = 1$ holds. □

References

[1] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph convolutional matrix completion. arXiv:1706.02263. Retrieved from https://arxiv.org/abs/1706.02263

[2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American Statistical Association* 112, 518 (2017), 859–877.

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 104–112.

[4] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. Autodebias: Learning to debias for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 21–30.

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. arXiv:2010.03240. Retrieved from https://doi.org/10.48550/arXiv.2010.03240

[6] Jiawei Chen, Yan Feng, Martin Ester, Sheng Zhou, Chun Chen, and Can Wang. 2018. Modeling users' exposure with social knowledge influence and consumption influence for recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 953–962.

[7] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. 2020b. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. In *Proceedings of the AAAI Conference on artificial intelligence*, Vol. 34. 3470–3477.

[8] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2019b. Samwalker: Social recommendation with informative sampling strategy. In *The World Wide Web Conference*. 228–239.

[9] Lei Chen, Le Wu, Kun Zhang, Richang Hong, Defu Lian, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Improving recommendation fairness via data augmentation. In *Proceedings of the ACM Web Conference 2023*. 1012–1020.

[10] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 456–464.

[11] Sirui Chen, Jiawei Chen, Sheng Zhou, Bohao Wang, Shen Han, Chanfei Su, Yuqing Yuan, and Can Wang. 2024. SIGformer: Sign-aware graph transformer for recommendation. *In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24).* ACM, New York, NY, 1274–1284. DOI: https://doi.org/10.1145/3626772.3657747

[12] Andrew Collins, Dominika Tkaczyk, Akiko Aizawa, and Joeran Beel. 2018. A study of position bias in digital library recommender systems. arXiv:1802.06565. Retrieved from https://arxiv.org/abs/1802.06565

[13] Yu Cui, Feng Liu, Pengbo Wang, Bohao Wang, Heng Tang, Yi Wan, Jun Wang, and Jiawei Chen. 2024. Distillation matters: Empowering sequential recommenders to match the performance of large language model. arXiv:2405.00338. Retrieved from https://arxiv.org/abs/2405.00338

[14] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B* (Methodological) 39, 1 (1977), 1–22.

[15] Leyan Deng, Defu Lian, Chenwang Wu, and Enhong Chen. 2022. Graph convolution network based recommender systems: Learning guarantee and item mixture powered strategy. *Advances in Neural Information Processing Systems* 35 (2022), 3900–3912.

[16] Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. 2021. On the equivalence of decoupled graph convolution network and label propagation. In *Proceedings of the Web Conference 2021*. 3651–3662.

[17] Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 238–248.

[18] Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. 2022. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems* (2022).

[19] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[20] Jiayan Guo, Peiyan Zhang, Chaozhuo Li, Xing Xie, Yan Zhang, and Sunghun Kim. 2022. Evolutionary preference learning via graph nested GRU ODE for session-based recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 624–634.

[21] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 639–648.

[22] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2227–2233.

[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] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 549–558.

[25] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *International Conference on Machine Learning*. PMLR, 1512–1520.

[26] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. arXiv:1511.06939. Retrieved from https://doi.org/10.48550/arXiv.1511.06939

[27] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of the 26th International Conference on World Wide Web*. 193–201.

[28] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*. IEEE, 263–272.

[29] Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, and Yongfeng Zhang. 2023. UP5: Unbiased foundation model for fairness-aware recommendation. arXiv:2305.12090. Retrieved from https://arxiv.org/abs/2305.12090

[30] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately interpreting clickthrough data as implicit feedback. *ACM SIGIR Forum* 51, 1 (2017), 4–11.

[31] Christopher C. Johnson. 2014. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems* 27, 78 (2014), 1–9.

[32] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[33] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[34] Jae-woong Lee, Seongmin Park, and Jongwuk Lee. 2021. Dual unbiased recommender learning for implicit feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1647–1651.

[35] Jae-woong Lee, Seongmin Park, Joonseok Lee, and Jongwuk Lee. 2022. Bilateral self-unbiased learning from biased implicit feedback. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 29–39. DOI: https://doi.org/10.1145/3477495.3531946

[36] Qian Li, Xiangmeng Wang, Zhichao Wang, and Guandong Xu. 2023. Be causal: De-biasing social network confounding in recommendation. *ACM Transactions on Knowledge Discovery from Data* 17, 1 (2023), 1–23.

[37] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2022. Fairness in recommendation: A survey. arXiv:2205.13619. Retrieved from https://doi.org/10.48550/arXiv.2205.13619

[38] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on World Wide Web*. 951–961.

[39] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web (WWW '18)*. Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.), ACM, New York, 689–698. DOI: https://doi.org/10.1145/3178876.3186150

[40] Siyi Lin, Chongming Gao, Jiawei Chen, Sheng Zhou, Binbin Hu, and Can Wang. 2024. How do recommendation models amplify popularity bias? An analysis from the spectral perspective. arXiv:2404.12008. Retrieved from https://doi.org/10.48550/arXiv.2404.12008

[41] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 831–840.

[42] Yiming Liu, Xuezhi Cao, and Yong Yu. 2016. Are you influenced by others when rating? Improve rating prediction by conformity modeling. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 269–272.

[43] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems 2019 (NeurIPS '19)*. Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.), 5712–5723. DOI: https://proceedings.neurips.cc/paper/2019/hash/a2186aa7c086b46ad4e8bf81e2a3a19b-Abstract.html

[44] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative filtering and the missing at random assumption. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*. 267–275.

[45] Maeve O'Brien and Mark T. Keane. 2006. Modeling result-list searching in the World Wide Web: The role of relevance topologies and trust bias. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, Vol. 28, Citeseer, 1881–1886.

[46] Rong Pan and Martin Scholz. 2009. Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 667–676.

[47] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *Proceedings of the 8th IEEE International Conference on Data Mining*. IEEE, 502–511.

[48] 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2359–2367.

[49] Ruihong Qiu, Sen Wang, Zhi Chen, Hongzhi Yin, and Zi Huang. 2021. CausalRec: Causal inference for visual debiasing in visually-aware recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3844–3852.

[50] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618. Retrieved from https://doi.org/10.48550/arXiv.1205.2618

[51] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. 811–820.

[52] Yuta Saito. 2020. Unbiased pairwise learning from biased implicit feedback. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 5–12.

[53] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[54] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. 111–112.

[55] Harald Steck. 2019. Embarrassingly shallow autoencoders for sparse data. In *Proceedings of the the World Wide Web Conference (WWW '19)*. Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.), ACM, New York, NY, 3251–3257. DOI: https://doi.org/10.1145/3308558.3313710

[56] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*. 729–739.

[57] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748. Retrieved from http://arxiv.org/abs/1807.03748

[58] Bohao Wang, Jiawei Chen, Changdong Li, Sheng Zhou, Qihao Shi, Yang Gao, Yan Feng, Chun Chen, and Can Wang. 2024. Distributionally robust graph-based recommendation system. In *Proceedings of the ACM on Web Conference 2024*. 3777–3788.

[59] Hao Wang, Yao Xu, Cheng Yang, Chuan Shi, Xin Li, Ning Guo, and Zhiyuan Liu. 2023. Knowledge-adaptive contrastive learning for recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. 535–543.

[60] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. 2018. Collaborative filtering with social exposure: A modular approach to social recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[61] Ting Wang and Dashun Wang. 2014. Why Amazon's ratings might mislead you: The story of herding effects. *Big Data* 2, 4 (2014), 196–204.

[62] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, New York, NY, 165–174. DOI: https://doi.org/10.1145/3331184.3331267

[63] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *Proceedings of the the World Wide Web Conference (WWW '19)*. ACM, New York, NY, 2022–2032. DOI: https://doi.org/10.1145/3308558.3313562

[64] 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).

[65] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Xiang Wang, and Xiangnan He. 2024. Understanding contrastive learning via distributionally robust optimization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

[66] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2020b. Self-supervised graph learning for recommendation. arXiv:2010.10783. Retrieved from https://arxiv.org/abs/2010.10783

[67] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 4425–4445.

[68] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.

[69] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 279–287.

[70] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. 2023. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*. 1063–1073.

[71] Dianer Yu, Qian Li, Xiangmeng Wang, and Guandong Xu. 2023a. Deconfounded recommendation via causal intervention. *Neurocomputing* 529 (2023), 128–139.

[72] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary?: Simple graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.), ACM, New York, NY, 1294–1303. DOI: https://doi.org/10.1145/3477495.3531937

[73] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2023. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering* 36, 1 (2023), 335–355.

[74] 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 *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 11–20.

[75] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.

[76] Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. 2020. Unbiased implicit recommendation and propensity estimation via combinational joint learning. In *Fourteenth ACM Conference on Recommender Systems*. 551–556.

Received 9 July 2023; revised 27 May 2024; accepted 3 June 2024