Representation Matters When Learning From Biased Feedback in Recommendation

Teng Xiao *
The Pennsylvania State University
tengxiao@psu.edu

Zhengyu Chen *
Zhejiang University
chenzhengyu@zju.edu.cn

Suhang Wang
The Pennsylvania State University
szw494@psu.edu

ABSTRACT

The logged feedback for training recommender systems is usually subject to selection bias, which could not reflect real user preference. Thus, many efforts have been made to learn the de-biased recommender system from biased feedback. However, existing methods for dealing with selection bias are usually affected by the error of propensity weight estimation, have high variance, or assume access to uniform data, which is expensive to be collected in practice. In this work, we address these issues by proposing Learning De-biased Representations (LDR), a framework derived from the representation learning perspective. LDR bridges the gap between propensity weight estimation (WE) and unbiased weighted learning (WL) and provides an end-to-end solution that iteratively conducts WE and WL. We show LDR can effectively alleviate selection bias with bounded variance. We also perform theoretical analysis on the statistical properties of LDR, such as its bias, variance, and generalization performance. Extensive experiments on both semi-synthetic and real-world datasets demonstrate the effectiveness of LDR.

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

unbiased learning; logged feedback; counterfactual learning

ACM Reference Format:
Teng Xiao, Zhengyu Chen, and Suhang Wang . 2022. Representation Matters When Learning From Biased Feedback in Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557431

1 INTRODUCTION

Recommender systems (RS) aim to infer user preferences from logged feedback and recommend items that users might like. The ideal logged feedback should be collected by randomly and uniformly exposing items to users [6, 36, 37, 50]. However, due to the feedback loop in RS, the exposures are affected by some underlying mechanisms, such as the past recommendation policy and user-selection. Thus, the feedback is missing-not-at-random (MNAR) [36, 37]. As shown in Figure 1, if the observed feedback is collected under the most popular policy, i.e., a policy that always recommends popular items to all users, the probability of exposure for popular items may be large. Thus, selection bias can exacerbate popularity bias, causing not relevant but popular items to be shown. In addition, due to user self-selection, users tend to give feedback to items they like [34, 37]. Thus, the observed logged feedback can be substantially higher than those not observed [37]. Previous studies [34, 37] also have shown that directly learning from biased feedback will lead to a biased estimation of users' true preferences. To address the selection bias issue, many efforts have been taken [4–6, 36, 37]. Among them, causal inference methods such as inverse propensity score (IPS) [36, 37] and doubly robust (DR) methods [35, 43] come with strong theoretical insights. Nevertheless, we find these methods have several limitations. First, accurately estimating the propensity score is critical for these methods; while correctly estimating propensity score is typically very difficult as model misspecification often occurs in real-world settings [34]. Second, those methods essentially follow a pipelined two-step paradigm as shown in Figure 2 (a): (1) conducting the weight estimation (WE); (2) using the estimated weights to do the unbiased weighted learning (WL). However, the WE process completely disregards the need to improve the performance of unbiased WL. Thus they may not give the optimal solution since the unbiased WL performance is sensitive to the pre-estimated weight and there is a gap between WE and WL due to the divergence of optimization objectives in the two separated stages. Third, it is shown that IPS-based methods have large variance [36, 37], especially when there exists severe selection bias with the large item space.

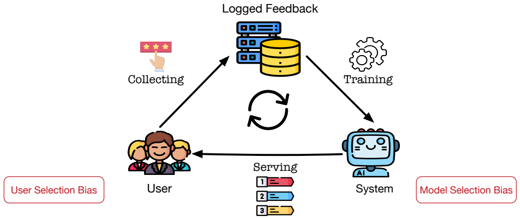


Figure 1: The closed feedback loop in RS. *Model Selection Bias*: The exposure of items in the serving stage is not uniform but is affected by the previous systems, which control what items to show. *User Selection Bias*: users may click some items more than others (even with a bias-free random recommender), which means that a few items receive more feedback while the majority have fewer ones.

More recently, several works try to address selection bias with various machine learning methods such as meta-learning [5, 44], domain adaptation [4], knowledge distillation [24], information bottleneck [25, 45]. Despite their promising performance, in addition to MNAR feedbacks, most of them [4, 5, 23, 24, 44] need

[*] Equal Contributions

CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557431

unbiased uniform feedback, i.e., feedbacks collected by randomly displaying items to users; while collecting uniform feedback is expensive and impractical in real-world as it hurts users' experiences and might cause significantly loss for domains like RS in healthcare. In addition, these methods [4, 24, 25, 45] do not have strong theoretical justification for their unbiasedness and estimation variance compared to causal inference-based methods, e.g., IPS and DR. Motivated by the discussion above, in this paper, we investigate whether one can effectively address the selection bias issue without any unbiased uniform feedback while still theoretically quantifying the trade-off between the estimation bias and variance. Our key idea is to alleviate the divergence between WE and unbiased WL in the causal inference methods. However, alleviating this divergence is non-trivial, and we are faced with two main challenges: (i) How to bridge the gap caused by different optimization objectives? Existing causal inference strategies for unbiased learning fall into a two-stage paradigm. The optimization gap between the two steps significantly limits their ability to generalize to downstream unbiased performance; and (ii) How to achieve a good bias-variance trade-off and theoretically guarantee the generalization performance when we alleviate the divergence between WE and unbiased WL steps?

To address these challenges, we propose an effective framework named Learning De-biased Representations (LDR). Specifically, for the first challenge, LDR learns de-biased representations of user-item pairs by simultaneously conducting WE and WL in an end-to-end process as shown in Figure 2 (b). For the second challenge, an adversarial discriminator is trained with representation learning to effectively bound the estimation variance. We also note that the adversarial representation learning has been applied broadly in fairness [13], causal inference [20] and domain adaptation [15]. In contrast, in this paper, we adopt adversarial representation learning on the unbiased recommendation, demonstrating that minimax representation learning is effective for reducing estimation variance. Technically, the key differences of our framework from them [13, 15, 20, 26] are that they do not weight/reweight source risk and do not consider the bias and variance. LDR joints strength from representation learning, weight estimation and representation adaptation, resulting in a principled framework that better addresses the challenges for unbiased recommendation. We provide theoretical guarantees for LDR and quantify the trade-off between the bias and variance. The main contributions of this research are:

- We propose a principled learning framework (LDR), which can alleviate the divergence between WE and unbiased WL objectives and shed a new representation learning perspective on the unbiased recommendation for the first time;
- We theoretically analyze the statistical properties and show that our LDR framework can achieve the unbiased estimation with bounded variance and have better generalization performance;
- Extensive experiments on both semi-synthetic and real-world datasets show that our LDR can outperform existing unbiased algorithms in the presence of selection bias for recommendation.

2 RELATED WORK

Selection Bias Correction. Selection bias occurs when a data sample is not representative of the underlying data distribution. To alleviate the selection bias in recommendation, inverse propensity score (IPS)-based methods [36, 37, 54] from causal inference [10, 40, 51] are adopted. The doubly robust (DR) methods [43, 44] further combine the propensity score estimation and the error imputation model to reduce the variance of IPS. Although these IPS and DR methods can in theory get an unbiased model by reweighting each sample, they heavily rely on the quality of the data imputation model or the propensity estimation model; while it is impossible to know the true propensity score or imputation model. In addition, previous works [12, 41] have shown that the propensity-based estimators suffer from very large variance issue [33, 41].

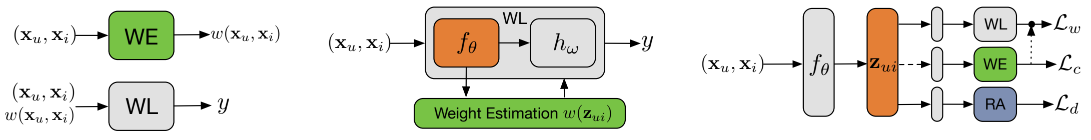
To avoid estimating the propensity score, some recent methods have been proposed and they are inspired by various machine learning techniques such as meta learning [5, 44], domain adaptation [4], knowledge distillation [24], and transfer learning [23]. The high-level idea for these methods is utilizing uniform data to guide the learning of debiasing parameters. Although these methods achieve promising performance, collecting the uniform data is extraordinarily expensive in practice. While some debiasing methods [25, 45, 50] do not assume access to uniform data, they lack theoretical unbiasedness and variance guarantees. In this paper, we focus on developing a theoretically unbiased learning framework to deal with selection bias in recommendation without any uniform data. Our insight that iteratively conducting WE and unbiased WL without any uniform data is different from the above methods.

The selection bias issue also comes up in other areas, such as off-policy learning [46, 49] and counterfactual learning [17, 20, 21, 48, 52]. However, off-policy learning operates on interactive logs and focuses on maximizing the reward and counterfactual learning focus on the causal effect estimation, which are different from the unbiased ranking task that we consider.

Adversarial Representation Learning. Our work is also related to but *different* from current adversarial representation learning methods. The adversarial formulation has been applied broadly in fairness [13], unsupervised domain adaptation [15], and causal inference [8, 20]. We formulate this on unbiased recommendation, and to demonstrate that minimax optimization is effective for solving the selection bias, both theoretically and empirically. Among the methods mentioned above, the unsupervised domain adaptation (UDA) methods called domain-invariant feature learning [1, 27] are most similar to ours. Our approach further develops this approach to avoid the high estimation variance. The main difference between UDA from ours is that UDA methods do not deal with missing data and associated challenges and do not consider the technique of re-weighting, while we are interested in re-weighting the objective function to alleviate selection bias with lower variance.

3 PRELIMINARIES

Let $\mathcal{U}$ be a set of users, and $\mathcal{I}$ be a set of items. $\mathcal{X}_u$ and $\mathcal{X}_i$ are two feature spaces of dimensions d_u and d_i, respectively. We use $\mathbf{x}_u \in \mathcal{X}_u$ and $\mathbf{x}_i \in \mathcal{X}_i$ to denote the features of a user u and an item i, respectively. Typically, $\mathbf{x}_u$ and $\mathbf{x}_i$ are the one-hot encodings of user and item IDs, respectively. The objective for recommendation is to estimate a parametric function $h_\omega(\mathbf{x}_u, \mathbf{x}_i) : \mathcal{X}_u \times \mathcal{X}_i \rightarrow \mathcal{Y}$ that maps the user and item features to a feedback $y \in \mathcal{Y}$, where ω denotes the learning parameters of h. Generally, we are given an


$(\mathbf{x}_u, \mathbf{x}_i)$
WE
$w(\mathbf{x}_u, \mathbf{x}_i)$
$(\mathbf{x}_u, \mathbf{x}_i)$
$w(\mathbf{x}_u, \mathbf{x}_i)$
WL
y
$(\mathbf{x}_u, \mathbf{x}_i)$
WL
f_θ
h_ω
y
Weight Estimation $w(\mathbf{z}_{ui})$
$(\mathbf{x}_u, \mathbf{x}_i)$
f_θ
$\mathbf{z}_{ui}$
WL
WE
RA
$\mathcal{L}_w$
$\mathcal{L}_c$
$\mathcal{L}_d$

(a) Static Weighted Learning (b) Iterative Weighted Learning (c) Network Architecture

Figure 2: (a) The pipelined and static process in IPS and DR, which firstly conducts weight estimation (WE) and then plugs it into the weighted learning (WL). (b) The circle and iterative process in our LDR (c) Proposed LDR for unbiased learning under selection bias. RA denotes proposed representation adaptation, and dashed lines are not back-propagated through.

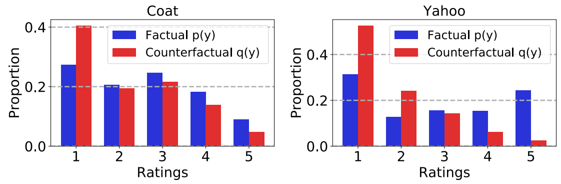


Figure 3: Rating distributions of factual (training) $p(y)$ and counterfactual (testing) $q(y)$ on Yahoo! R3 and Coat datasets.

observational dataset $\mathcal{D}_p$ of N triples of user, item, and feedback, i.e., $\mathcal{D}_p \triangleq \{x_u^{(n)}, x_i^{(n)}, y^{(n)}\}_{n=1}^{N}$, and our task is to learn h_ω with $\mathcal{D}_p$. For brevity, we drop the superscript n in what follows.

3.1 Unbiased Recommendation

Generally, the observational samples in $\mathcal{D}_p$ suffer from selection bias due to various issues, e.g., the set of items exposed to users is affected by the past recommendation policy and users tend to give feedback to items they like. Thus, we can treat $(\mathbf{x}_u, \mathbf{x}_i, y) \in \mathcal{D}_p$ as sampled from a generative process that depends on user self-selection and past recommendation policy [37], i.e., $p(\mathbf{x}_u, \mathbf{x}_i, y) = p(\mathbf{x}_u)p(\mathbf{x}_i|\mathbf{x}_u)p(y|\mathbf{x}_u, \mathbf{x}_i)$, where $p(\mathbf{x}_u)$ is the uniform distribution of users. $p(\mathbf{x}_i|\mathbf{x}_u)$ depends on user self-selection along with the underlying past recommendation policy, which is also called the exposure probability $p(O_{ui})$ in [36, 37, 43] and is unknown ahead of time. Ideally, we are interested in learning unbiased h_ω with ideal risk function under uniform exposure distributions as follows:

$$\mathcal{L}_{ideal}(\omega) = \mathbb{E}_{q(\mathbf{x}_u,\mathbf{x}_i,y)}[\ell(h_\omega(\mathbf{x}_u, \mathbf{x}_i), y)], \tag{1}$$

where $q(\mathbf{x}_u, \mathbf{x}_i, y) = p(\mathbf{x}_u)p(\mathbf{x}_i)p(y|\mathbf{x}_u, \mathbf{x}_i)$ is the counterfactual distribution and $\ell(.)$ is the loss associated with each sample pair. Note that the ideal loss function is independent of the conditional distribution $p(\mathbf{x}_i|\mathbf{x}_u)$. That is, we calculate the sample-wise loss over the marginal uniform distributions of users and items, $p(\mathbf{x}_u) = \frac{1}{|\mathcal{U}|}$ and $p(\mathbf{x}_i) = \frac{1}{|\mathcal{I}|}$ and feedbacks, rather than their joint distribution $p(\mathbf{x}_u, \mathbf{x}_i) = p(\mathbf{x}_u)p(\mathbf{x}_i|\mathbf{x}_u)$. This is because de-biasing recommender aims to predict feedback under alternative matches different from the ones observed in the MNAR data. In other words, *we want our estimated h_ω to generalize well for all possible pairs of users and items, not just the pairs that are frequently matched in the observational data.* However, the empirical estimate of true risk in Eq. (1) is inaccessible as we only can use the MNAR data $\mathcal{D}_p$ for the empirical estimation. Directly learning h_ω with standard supervised loss on $\mathcal{D}_p$ could suffer from selection bias due to the discrepancy between the factual distribution $p(\mathbf{x}_u, \mathbf{x}_i, y) = p(\mathbf{x}_u)p(\mathbf{x}_i|\mathbf{x}_u)p(y|\mathbf{x}_u, \mathbf{x}_i)$ in the training MNAR data, and the counterfactual testing distribution $q(\mathbf{x}_u, \mathbf{x}_i, y) = p(\mathbf{x}_u)p(\mathbf{x}_i)p(y|\mathbf{x}_u, \mathbf{x}_i)$ to which the model will be practically applied. To verify it, we plot the marginal distributions of $p(y)$ and $q(y)$ on two datasets for unbiased recommendation [28, 37] in Figure 3, which shows significant difference between $p(y)$ and $q(y)$. Mathematically, under this distribution discrepancy, standard empirical risk minimization $\mathcal{L}_{sl}$ over data $\mathcal{D}_p$ is not an unbiased estimate of the true risk $\mathcal{L}_{ideal}(\omega)$ [37]:

$$\begin{aligned} &\hat{\mathcal{L}}_{sl}(\omega) \simeq \mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\ell(h_\omega(\mathbf{x}_u, \mathbf{x}_i), y)] \neq \mathcal{L}_{ideal}(\omega), \\ &\text{where } \hat{\mathcal{L}}_{sl}(\omega) = \frac{1}{N}\sum\nolimits_{(\mathbf{x}_u,\mathbf{x}_i,y)\in\mathcal{D}_p} \ell(h_\omega(\mathbf{x}_u, \mathbf{x}_i), y). \end{aligned} \tag{2}$$

Problem Definition: With the definitions above, the studied problem can be defined as: *Given only MNAR dataset $\mathcal{D}_p$, build an unbiased estimator for the ideal loss and learn the parameterized function $h_\omega(\mathbf{x}_u, \mathbf{x}_i)$ to improve unbiased recommendation performance.*

4 LEARNING DE-BIASED REPRESENTATION

As discussed in § 3, directly learning $h_\omega(\mathbf{x}_u, \mathbf{x}_i)$ via the empirical risk minimization is infeasible as the empirical risk in Eq. (1) is inaccessible. To address this challenge, we propose a new learning framework. The key idea of our framework is iteratively conducting WE and unbiased WL in a seamless manner (see Fig. 2 (b)), and alleviating the variance induced in the latent representation space.

An illustration of our LDR is shown in Figure 2 (c), which is composed of an embedding function $f_\theta(\mathbf{x}_u, \mathbf{x}_i) = \mathbf{z}_{ui}$, a weighted learning (WL) component, a representation adaptation (RA) component and a weight estimation (WE) component. The embedding function f_θ maps samples into a latent representation space. WE takes the representation as input to estimate the sample weights, which is treated as density ratio to alleviate selection bias. RA adapts the indistinguishable representations to alleviate the learning variance. WL is for the final unbiased prediction of recommendation. Next, we give the details of LDR and theoretically show that it can conduct unbiased estimation without knowing propensity scores and has low variance dynamically.

4.1 De-biased Weighted Learning

As mentioned above, the main challenge for unbiased recommendation with selection bias is that the ideal risk function $\mathcal{L}_{ideal}(\omega) = \mathbb{E}_{q(\mathbf{x}_u,\mathbf{x}_i,y)}[\ell(h_\omega(\mathbf{x}_u, \mathbf{x}_i), y)]$ is defined over the target distribution $q(\mathbf{x}_u, \mathbf{x}_i, y)$ where the counterfactual feedback y is not observed. Even through Monte Carlo sampling, we cannot directly estimate $\mathcal{L}_{ideal}(\omega)$ via the empirical risk under $q(\mathbf{x}_u, \mathbf{x}_i, y)$. Thus, MF-IPS [37] adopts inverse propensity weighting:

$$\begin{aligned} \mathcal{L}_{ideal}(\omega) &= \mathbb{E}_{q(\mathbf{x}_u,\mathbf{x}_i,y)}[\ell(h_\omega(\mathbf{x}_u, \mathbf{x}_i), y)] \\ &= \int \frac{q(\mathbf{x}_u, \mathbf{x}_i, y)}{p(\mathbf{x}_u, \mathbf{x}_i, y)} \ell(h_\omega(\mathbf{x}_u, \mathbf{x}_i), y) p(\mathbf{x}_u, \mathbf{x}_i, y) d\mathbf{x}_u d\mathbf{x}_i dy \\ &= \mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\frac{q(\mathbf{x}_u, \mathbf{x}_i)}{p(\mathbf{x}_u, \mathbf{x}_i)} \ell(h_\omega(\mathbf{x}_i, \mathbf{x}_i), y)] = \mathcal{L}_{ips}(\omega), \end{aligned} \tag{3}$$

where $\mathcal{L}_{ips}(\omega)$ is an unbiased estimator of the ideal risk $\mathcal{L}_{ideal}(\omega)$. Note that DR [43] is also built on inverse propensity weighting but has an additional imputation model. Without loss of generality, we focus on IPS in this paper. Then, along with weight $\frac{q(\mathbf{x}_u,\mathbf{x}_i)}{p(\mathbf{x}_u,\mathbf{x}_i)} \triangleq w(\mathbf{x}_u,\mathbf{x}_i)$, an unbiased estimator of $\mathcal{L}_{ips}(h)$ can be obtained by reweighting the empirical risk under MNAR dataset $\mathcal{D}_p$ as follows:

$$\hat{\mathcal{L}}_{ips}(\omega) = \frac{1}{N} \sum_{(\mathbf{x}_u,\mathbf{x}_i,y)\in\mathcal{D}_p} w(\mathbf{x}_u,\mathbf{x}_i)\ell(h_\omega(\mathbf{x}_u,\mathbf{x}_i), y). \tag{4}$$

This idea has been applied to many algorithms [36, 37, 43, 44, 48, 54]. However, there are two limitations: (1) they all follow a two-step pipe-lined process (see Figure 2 (a)), which firstly estimates propensity weights and then plugs them into the model learning. In this two-step paradigm, the WE step is decoupled from the WL step. In particular, $w(\mathbf{x}_u, \mathbf{x}_i)$ is estimated without accommodating any form of adaptation that is potentially useful for future unbiased learning on the recommendation task. The apparent divergence between the two steps would result in sub-optimal unbiased learning (2) The estimated importance weight $w(\mathbf{x}_u, \mathbf{x}_i)$ can be very large, resulting in a large variance and sub-optimal estimation [41, 43, 47].

4.2 Representation Learning

To address the limitations mentioned above, we firstly propose to leverage an embedding function to project the high-dimensional input $(\mathbf{x}_u, \mathbf{x}_i)$ into a lower-dimensional representation space $\mathcal{Z}$. Our key insights are: (1) Dynamically embedding unbiased learning into the end-to-end process of representation learning can make the final representations both discriminative and unbiased. As a consequence, this more advanced the end-to-end solution can gradually improve weight estimation and reduce the bias of model learning in a seamless manner; (2) By mapping regions of low density in $(\mathcal{X}_u, \mathcal{X}_i)$ into regions of higher density in $\mathcal{Z}$, the representations are made more compact, and we expect that the weight estimation will be much easier; and (3) it paves us a way to further reduce the estimation variance via the representation adaptation in § 4.3.

Specifically, we apply a transformation of data rather than directly model $w(\mathbf{x}_u, \mathbf{x}_i)$ like IPS. Let $f_\theta : \mathcal{X}_u \times \mathcal{X}_i \longrightarrow \mathcal{Z} \in \mathbb{R}^{d_z}$ be the transformation function, where d_z is the reduced dimension with $d_z \ll d_u$ and d_i. Then $\mathbf{z}_{ui} = f_\theta(\mathbf{x}_u, \mathbf{x}_i)$ is the transformed random variable, whose randomness comes from $(\mathbf{x}_u, \mathbf{x}_i)$ exclusively. Given this transformation, we can estimate the weight $w(\mathbf{x}_u, \mathbf{x}_i)$ on the latent space. Specifically, the feasibility of applying the latent weight estimation can be proved by the following theory:

Theorem 4.1. *Given an invertible and deterministic mapping f_θ : $(\mathbf{x}_u, \mathbf{x}_i) \mapsto \mathbf{z}_{ui}$, let $p(\mathbf{z}_{ui})$ and $q(\mathbf{z}_{ui})$ be the probability density functions induced by $p(\mathbf{x}_u, \mathbf{x}_i)$, $q(\mathbf{x}_u, \mathbf{x}_i)$, and f_θ. Then we have*

$$w(\mathbf{x}_u,\mathbf{x}_i) = \frac{q(\mathbf{x}_u,\mathbf{x}_i)}{p(\mathbf{x}_u,\mathbf{x}_i)} = \frac{q(f_\theta(\mathbf{x}_u,\mathbf{x}_i))}{p(f_\theta(\mathbf{x}_u,\mathbf{x}_i))} = \frac{q(\mathbf{z}_{ui})}{p(\mathbf{z}_{ui})} = w(\mathbf{z}_{ui}). \tag{5}$$

We provide the proof in Appendix A.1. Theorem 4.1 contains our preliminary study of when and why the representation learning is expected to work. This theorem shows that, for any deterministic and invertible mapping $f_\theta(\mathbf{x}_u, \mathbf{x}_i)$, we can utilize the latent representation $\mathbf{z}_{ui}$ to conduct the weight estimation step. Note that invertible and deterministic are common and widely used assumptions in the literature [38, 39, 42] as a basic condition for analysis. Importantly, ensuring deterministic and invertibility is feasible for many recommendation backbones such as the matrix factorization collaborative filtering (MCF) [7, 9, 22, 31] and neural collaborative filtering (NCF) [18] with user and item one-hot embeddings. We empirically demonstrate that representation learning governs the success of our methods on MCF and NCF in § 5. Given the representation learning, the unbiased estimator in Eq. (4) now can be learned on the latent representation space $\mathcal{Z}$ as follows:

$$\begin{aligned}\mathcal{L}_w(\omega,\theta) &= \mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y),\ \mathbf{z}_{ui}=f_\theta(\mathbf{x}_u,\mathbf{x}_i)}\left[w(\mathbf{z}_{ui})\ell(h_\omega(\mathbf{z}_{ui}), y)\right] \\ &\simeq \frac{1}{N}\sum_{(\mathbf{x}_u,\mathbf{x}_i,y)\in\mathcal{D}_p} \widehat{w}\left(\mathbf{z}_{ui}\right)\ell\left(h_\omega\left(f_\theta(\mathbf{x}_u,\mathbf{x}_i)\right), y\right) = \widehat{\mathcal{L}}_w(\omega,\theta),\end{aligned} \tag{6}$$

where $\widehat{w}(\mathbf{z}_{ui}) = \widehat{w}(f_{\bar{\theta}}(\mathbf{x}_u, \mathbf{x}_i))$ is the empirical estimated weight. Thus, different from IPS and DR [36, 37, 43], optimizing this objective enables us to conduct the unbiased weighted learning in a dynamic and seamless way: at the t-th iteration. after $\widehat{w}(\mathbf{z}_{ui}) = \widehat{w}(f_{\bar{\theta}_t}(\mathbf{x}_u, \mathbf{x}_i))$ is estimated, θ_t will be updated to θ_{t+1} by optimizing Eq. (6), and the current $\bar{\theta}_t$ will move to the next $\bar{\theta}_{t+1}$; then, we estimate a new set of weights $\widehat{w}(\mathbf{z}_{ui}) = \widehat{w}(f_{\bar{\theta}_{t+1}}(\mathbf{x}_u, \mathbf{x}_i))$. A key contribution of our work is exactly this dynamic interaction in the training processes of weighted learning and weight estimation.

4.3 Weight Estimation

In the last subsection, we have demonstrated the importance of representation learning. Thus, with the optimization problem of Eq. (6), we expect the interaction in the training can boost the performance of both weighted learning and weight estimation. However, we do not know the optimal weight $\widehat{w}(\mathbf{z}_{ui})$ on the latent space. Therefore, another challenge for weight estimation is how to design an practical algorithm. We address this by first giving the following theory of the weight estimation on the latent representation space.

Theorem 4.2. *Given f_θ : $(\mathbf{x}_u, \mathbf{x}_i) \mapsto \mathbf{z}_{ui}$. Let $p(\mathbf{z}_{ui})$ and $q(\mathbf{z}_{ui})$ be the densities induced by $p(\mathbf{x}_u, \mathbf{x}_i)$, $q(\mathbf{x}_u, \mathbf{x}_i)$, and f_θ. Let $\mathcal{D}_q$ be n i.i.d pairs from $q(\mathbf{x}_u, \mathbf{x}_i) = \frac{1}{N} \cdot \frac{1}{M}$. If $\widehat{w}(\mathbf{x}_u, \mathbf{x}_i) = WE(\mathcal{D}_p, \mathcal{D}_q)$ is an empirical unbiased estimator for $w(\mathbf{x}_u, \mathbf{x}_i)$, then $\widehat{w}(\mathbf{z}_{ui}) = WE(f_\theta(\mathcal{D}_p), f_\theta(\mathcal{D}_q))$ is also an unbiased estimator for $w(\mathbf{x}_u, \mathbf{x}_i)$.*

We provide the proof in Appendix A.2. This theorem shows that we can estimate optimal weight $\widehat{w}(\mathbf{z}_{ui})$ in Eq. (6) by using the finite sample $\mathcal{D}_p$ drawn from $p(\mathbf{x}_u, \mathbf{x}_i)$ and $\mathcal{D}_q$ randomly drawn from $q(\mathbf{x}_u, \mathbf{x}_i)$ which is the known uniform distribution with $p(\mathbf{x}_u) = \frac{1}{|\mathcal{U}|}$ and $p(\mathbf{x}_i) = \frac{1}{|\mathcal{I}|}$. Note that we do not require the unbiased uniform feedback y in D_q and only need random unlabeled user-item pairs (x_u, x_i). In the following, we introduce the details of the weight estimation strategies. We adopt a discriminative weight estimation method [3], also known as the likelihood ratio trick, that has been applied across generative models [2] and reinforcement learning [14]. However, different from them, we provide theoretical analysis and analyze how estimated weights impact the bias and generalization performance. To get the empirical weight $\widehat{w}(\mathbf{z}_{ui})$, we use a learned binary classifier, which infers whether user-item pairs came from the factual distribution $p(\mathbf{x}_u, \mathbf{x}_i)$ or counterfactual $q(\mathbf{x}_u, \mathbf{x}_i)$. Specifically, we set the label of the data in $\mathcal{D}_q$ to be 0 and the label of the data in $\mathcal{D}_p$ to be 1, and fit a classifier $c_\phi(\mathbf{z}_{ui})$ by solving the following objective:

$$\begin{aligned}&\mathcal{L}_c(\phi) = \mathbb{E}_{\mathbf{z}_{ui}=f_\theta(\mathbf{x}_u,\mathbf{x}_i),(\mathbf{x}_u,\mathbf{x}_i)\sim p(\mathbf{x}_u,\mathbf{x}_i)}\left[\log\sigma(c_\phi(\mathbf{z}_{ui}))\right]+ \\ &\mathbb{E}_{\mathbf{z}_{ui}=f_\theta(\mathbf{x}_u,\mathbf{x}_i),(\mathbf{x}_u,\mathbf{x}_i)\sim q(\mathbf{x}_u,\mathbf{x}_i)}\left[\log\sigma(-c_\phi(\mathbf{z}_{ui}))\right] \\ &\simeq \frac{1}{N}\sum_{(\mathbf{x}_u,\mathbf{x}_i,y)\in\mathcal{D}_p}\left[\log\sigma(c_\phi(\mathbf{z}_{ui}))\right] + \frac{1}{N'}\sum_{(\mathbf{x}_u,\mathbf{x}_i)\in\mathcal{D}_q}\left[\log\sigma(-c_\phi(\mathbf{z}_{ui}))\right],\end{aligned} \tag{7}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ and $w(\mathbf{z}_{ui}) = w(f_{\bar{\theta}}(\mathbf{x}_u, \mathbf{x}_i))$. Given the optimized $c_\phi(\mathbf{z}_{ui})$, we can use Bayes' rule to get the empirical weight estimation $\widehat{w}(\mathbf{z}_{ui})$. The key idea is that probabilities $q(\mathbf{z}_{ui})$ and $p(\mathbf{z}_{ui})$ are related to the classifier probabilities via Bayes' rule:

$$\widehat{w}(\mathbf{z}_{ui}) = \frac{q(\mathbf{z}_{ui})}{p(\mathbf{z}_{ui})} = \frac{r(\mathbf{z}_{ui}|d=0)}{r(\mathbf{z}_{ui}|d=1)} = \frac{r(d=1)\widehat{r}(d=0|\mathbf{z}_{ui})}{r(d=0)\widehat{r}(d=1|\mathbf{z}_{ui})}, \tag{8}$$

where r is a distribution over $(\mathbf{z}_{ui}, d) \in \mathcal{Z} \times \{0,1\}$ and d denotes which world $\mathbf{z}_{ui}$ belongs. Given Eq. (8), the weight $\widehat{w}(\mathbf{z}_{ui})$ can be decomposed into two parts. The former $\frac{r(d=1)}{r(d=0)}$ is a constant and can be estimated with the sample sizes of factual and counterfactual worlds as $\frac{N}{|\mathcal{U}|\times|\mathcal{I}|}$. The second part $\frac{\widehat{r}(d=0|\mathbf{z}_{ui})}{\widehat{r}(d=1|\mathbf{z}_{ui})}$ is the ratio of counterfactual to factual that can be directly estimated with the probabilistic predictions of the logistic regression classifier: $\frac{\widehat{r}(d=0|\mathbf{z}_{ui})}{\widehat{r}(d=1|\mathbf{z}_{ui})} = \frac{1-\sigma(c_\phi(\mathbf{z}_{ui}))}{\sigma(c_\phi(\mathbf{z}_{ui}))}$. With the weights estimated in this way, we can conduct unbiased weighted learning in Eq. (6), and the bias of applying a classifier for empirical weight learning can be proved:

Theorem 4.3. *Let $w_m \geq 0$ be the maximum weight $\widehat{w}(\mathbf{z}_{ui})$ of any representation $\mathbf{z}_{ui}$. Then for any $\mathbf{z}_{ui}$ s.t. $P(d=1 \mid \mathbf{z}_{ui}) \neq 0$, the bias of the unbiased weighted learning is bounded by:*

$$\begin{aligned}|\mathcal{L}_{ideal}(\omega) - \widehat{\mathcal{L}}_w(\omega,\theta)| &\leq \frac{1}{2}\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(\ell(h_\omega(\mathbf{z}_{ui}), y)^2] \\ &+ \frac{1}{2}\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(w_m+1)^4(r(d=1|\mathbf{z}_{ui}) - \widehat{r}(d=1|\mathbf{z}_{ui}))^2].\end{aligned} \tag{9}$$

We provide the proof in Appendix A.3. Since the first term is not related to the classifier, we can focus on the second term. From this upper bound, we have two observations: (1) finding a good estimate $\widehat{r}(d=1|\mathbf{z}_{ui})$ for $r(d=1|\mathbf{z}_{ui})$ can effectively reduce the bias; (2) A smaller w_m leads to a smaller bias of the estimated unbiased loss.

4.4 Representation Adaptation

Through representation learning and weight estimation, we can make unbiased weighted learning more tractable. However, as mentioned earlier, the importance weights are not explicitly bounded, which might result in large variance [36, 37]. In addition, as shown in Theorem 4.3, bounded $w(\mathbf{z}_{ui})$, i.e., smaller w_m, can also lead to smaller bias. Thus, we first give the following upper bound for the variance of the learning in Eq. (6), which paves us a way to minimize the learning variance and bound the $w(\mathbf{z}_{ui})$.

Theorem 4.4. *Let $d_\alpha(q\|p) = 2^{D_\alpha(q\|p)} = (\int_x \frac{q(x)^\alpha}{p(x)^{\alpha-1}})^{\frac{1}{\alpha-1}}$ be the Rényi divergence [32] between p and q. Then, the variance of the unbiased weighted learning objective $\mathcal{L}_w(\omega,\theta)$ is bounded by:*

$$\begin{aligned}&\mathrm{Var}[\mathcal{L}_w] = \mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(\mathcal{L}_w)^2] - (\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\mathcal{L}_w])^2 \leq \\ &d_{\alpha+1}(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))(\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\mathcal{L}_w])^{\frac{\alpha-1}{\alpha}} - (\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\mathcal{L}_w])^2\end{aligned} \tag{10}$$

$\forall\alpha > 0$, where we denote $\mathcal{L}_w(\omega,\theta)$ by $\mathcal{L}_w$ for brevity.

We provide the proof in Appendix A.4. Apparently, the variance of the weighted estimator is bounded by Renyi divergence. However, it is challenging to reduce the Renyi divergence between $q(\mathbf{z}_{ui})$ and $p(\mathbf{z}_{ui})$ since we do not know the explicit density functions of them. To address this challenge, we propose to utilize the adversarial learning to reduce the divergence $d_{\alpha+1}(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))$. Since this upper bound holds for any $\alpha > 0$, without loss of generality, we focus on reducing $d_{\alpha+1}(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))$ with $\alpha = 1$. With some calculations, we have the following equation:

$$\begin{aligned}d_2(q(\mathbf{z})_{ui}\|p(\mathbf{z}_{ui})) &= \int_{\mathbf{z}_{ui}} \frac{q(\mathbf{z}_{ui})^2}{p(\mathbf{z}_{ui})} d\mathbf{z}_{ui} \\ = \int_{\mathbf{z}_{ui}} p(\mathbf{z}_{ui})[(\frac{q(\mathbf{z}_{ui})}{p(\mathbf{z}_{ui})})^2 - 1 + 1]d\mathbf{z}_{ui} &= D_s(q(\mathbf{z}_{ui}))\|p(\mathbf{z}_{ui})) + 1,\end{aligned} \tag{11}$$

where D_s is the f-divergence [11] with $s(t) = t^2 - 1$ being a convex function in the domain $\{t : t \geq 0\}$ and $s(1) = 0$. Given the equation above, we can focus on reducing $D_s(q(\mathbf{z}_{ui}))\|p(\mathbf{z}_{ui}))$. Inspired by the variational characterization of f-divergences [30], we estimate f-divergences from samples with variational optimization:

$$\begin{aligned}D_s(q(\mathbf{z}_{ui})||p(\mathbf{z}_{ui})) &= \int_{\mathbf{z}_{ui}} p(\mathbf{z}_{ui}) s(\frac{q(\mathbf{z}_{ui})}{p(\mathbf{z}_{ui})}) d\mathbf{z}_{ui} \\ &= \sup_T \{\mathbb{E}_{\mathbf{z}_{ui}\sim q(\mathbf{z}_{ui})}[T(\mathbf{z}_{ui})] - \mathbb{E}_{\mathbf{z}_{ui}\sim p(\mathbf{z}_{ui})}[s^*(T(\mathbf{z}_{ui}))]])\} \\ &\geq \sup_{T\in\mathcal{T}} \{\mathbb{E}_{\mathbf{z}_{ui}\sim q(\mathbf{z}_{ui})}[T(\mathbf{z}_{ui})] - \mathbb{E}_{\mathbf{z}_{ui}\sim p(\mathbf{z}_{ui})}[s^*(T(\mathbf{z}_{ui}))]\},\end{aligned} \tag{12}$$

where the second equality holds since s is a convex function and applying Fenchel convex duality ($s^*(y) := \sup_{x\in\mathbb{R}_+}\{xy - s(x)\} = y^2/4 + 1$) gives the dual formulation. The third inequality holds since we restrict T to a family of functions instead of all measurable functions. Fortunately, if we utilize the neural networks as the the family of T, the condition of this inequality can be satisfied due to the universal approximation theorem [19]. Specifically, we represent T as a discriminator $d_\psi(\mathbf{z}_{ui})$. We then view our feature extractor $f_\theta(\mathbf{x}_u, \mathbf{x}_i)$ as another generator neural network mapping $(\mathbf{x}_u, \mathbf{x}_i)$ to the probability of sampling $\mathbf{z}_{ui}$. Then, minimizing the f-divergence in Eq. (12) results in a min-max objective:

$$\begin{aligned}\mathcal{L}_d(\theta,\psi) &= \min_\theta \max_\psi \mathbb{E}_{\mathbf{z}_{ui}=f_\theta(\mathbf{x}_u,\mathbf{x}_i),(\mathbf{x}_u,\mathbf{x}_i)\sim q(\mathbf{x}_u,\mathbf{x}_i)}[d_\psi(\mathbf{z}_{ui})] \\ &- \mathbb{E}_{\mathbf{z}_{ui}=f_\theta(\mathbf{x}_u,\mathbf{x}_i),(\mathbf{x}_u,\mathbf{x}_i)\sim p(\mathbf{x}_u,\mathbf{x}_i)}[s^*(d_\psi(\mathbf{z}_{ui}))] \\ &\simeq \min_\theta \max_\psi \frac{1}{N}\sum_{(\mathbf{x}_u,\mathbf{x}_i,y)\in\mathcal{D}_p}[d_\psi(\mathbf{z}_{ui})] + \frac{1}{N'}\sum_{(\mathbf{x}_u,\mathbf{x}_i)\in\mathcal{D}_q}[s^*(d_\psi(\mathbf{z}_{ui}))].\end{aligned} \tag{13}$$

Intuitively, the objective of adversarial training makes the distribution counterfactual $q(\mathbf{z}_{ui})$ be closed to factual $p(\mathbf{z}_{ui})$, which results $w(\mathbf{z}_{ui}) = \frac{q(\mathbf{z}_{ui})}{p(\mathbf{z}_{ui})} \rightarrow 1$. The advantages of using this adversarial representation learning are two-folds: (i) It can reduce the bias since we have smaller w_m as shown by Theorem 4.3; (ii) It can reduce the variance since we bound Renyi divergence between $q(\mathbf{z}_{ui})$ and $p(\mathbf{z}_{ui})$ as shown in Theorem 4.4. To give more insights into why reducing Renyi divergence improves unbiased learning, we further provide the following generalization learning bound:

Theorem 4.5. *Let $w_m \geq 0$ be the maximum weight $\widehat{w}(\mathbf{z}_{ui})$ and $l_m \geq 0$ be the maximum value of per-sample loss ℓ (typically, $l_m = 1$ if we use log-loss). If $\widehat{w}(\mathbf{z}_{ui})$ is an unbiased estimation of $w(\mathbf{z}_{ui})$, then the following upper bound holds with probability at least $1-\delta$:*

$$\begin{aligned}&|\mathcal{L}_{ideal}(\omega) - \widehat{\mathcal{L}}_w(\omega,\theta)| \\ &\leq \frac{w_m l_m \log 1/\delta}{3N} + l_m\sqrt{\frac{2d_2(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))\log 1/\delta}{N}}.\end{aligned} \tag{14}$$

We provide the proof in Appendix A.5. Different from other generalization bounds [5, 34, 37, 43] in the unbiased recommendation, which is based on observation space and Hoeffding's inequality, our bound is based on the latent space and Bernstein inequality. This bound shows that, although the estimated weight is unbiased, the learning performance will still be bad if the divergence between

$q(\mathbf{z}_{ui})$ and $p(\mathbf{z}_{ui})$ is large. Theorem 4.3 and Theorem 4.5 both suggest that making a trade-off between the variance and bias can make $\widehat{\mathcal{L}}_w(\omega, \theta)$ be a more accurate estimation for $\mathcal{L}_{ideal}(\omega)$.

4.5 Final Objective Function

Based on the analysis above, we have h_ω for recommendation score prediction, classifier c_ϕ for weight estimation, d_ψ with adversarial learning to force the representations extracted by f_θ are bounded. Combining all these together, our final objective is:

$$\mathcal{L} = \min_{\omega,\phi,\theta} \max_{\psi} \mathcal{L}_w(\omega, \theta) + \alpha \mathcal{L}_c(\phi) + \gamma \mathcal{L}_d(\theta, \psi), \tag{15}$$

where α and γ are hyper-parameters to balance the contributions of classification loss and adversarial loss. An training algorithm is presented in Alg. 1 in Appendix. As shown in Figure 2 (c), we reuse the representation obtained from the backbone f_θ and just model our components (h_ω, c_ϕ and d_ψ) using three heads. Hence, compared with vanilla RS algorithms, the proposed LDR introduces few additional parameters and is efficient.

5 EXPERIMENT

In this section, we empirically evaluate the effectiveness of our LDR. Specifically, we answer the following questions.
(RQ1) How does the LDR perform compared with baselines?
(RQ2) How do different components affect the performance?
(RQ3) Can LDR leverage uniform data to tackle selection bias?
(RQ4) Do the proposed representation learning and adaptation work as designed and give some useful insights?

5.1 Experimental Setup

Dataset. To evaluate unbiased performance, we need biased and unbiased testing data collected by uniformly displaying items to users. We use two widely used real-world datasets which satisfy this requirement: `Yahoo!R3` [28] and `Coat` [37]. In addition, we also utilize a relatively large semi-synthetic dataset based on the Amazon-`Electronics` dataset [29]. Following previous works [6, 24, 45], we treat items rated greater than or equal to 3 as 1, and the others are considered as 0 for all datasets. Since `Electronics` does not contain an unbiased test set, following previous works [4, 34, 53], we simulate an unbiased test set where testing data are sampled by a uniform distribution over items with a skewed splitting. The details of datasets and the skewed splitting are given in Appendix B.
Model Architecture. For fair evaluation, we use NCF [18] as model architecture for all methods. The formulation of NCF is:

$$h_\omega^{\mathrm{NCF}}(u, i) := \mathrm{NN}(\omega; [\mathbf{z}_u, \mathbf{z}_i]), \tag{16}$$

where $\mathrm{NN}(\omega; \mathbf{x}) = \mathbf{W}_1 \tilde{\sigma}(\mathbf{W}_2 \mathbf{x}), \mathbf{W}_1 \in \mathbb{R}^{1\times d_1}, \mathbf{W}_2 \in \mathbb{R}^{d_1 \times 2d_z}$ with $\tilde{\sigma}(\cdot)$ being the ReLU activation. $\mathbf{z}_u$ and $\mathbf{z}_i$ are the user and item embeddings generated by $f_\theta(\mathbf{x}_u, \mathbf{x}_i)$. For classifier c_ϕ and discriminator d_ψ, we use two networks with $[\mathbf{z}_u, \mathbf{z}_i]$ as the input.
Metrics. Following previous works [4, 5, 45], we use Normalized Discounted Cumulative Gain@10 (NDCG@10), Area under the ROC Curve (AUC) and Negative Log-Likelihood (NLL) loss under the unbiased uniform test set as our evaluation metrics.
Settings. We select the best configuration of hyper-parameters for all baselines based on NLL on the validation set. For all methods, the hyper-parameter search spaces are: dropout {0.2, 0.4, 0.6}, learning rate {0.001, 0.005, 0.01}, L2 weight-decay {1e-3, 1e-4, 1e-5, 1e-6}, embedding dimension {16, 32, 64, 128, 256}. For LDR, we search α from {0.2, 0.4, 0.6, 0.8} and γ from {0.1, 1.0, 10, 100}.

5.2 (RQ1) Debiasing Performance Comparison

Baselines. In this section, we evaluate the debiasing performance under the scenario that we do not have unbiased uniform data in training. We compare with the following baselines: Vanilla (trained without any debiasing procedure), Inverse Propensity Score (IPS) with the normalization trick [37], Doubly Robust (DR) [43], Counterfactual Variational Information Bottleneck (CVIB) [45], Adversarial Counterfactual Learning (ACL) [50] and Asymmetric Tri-training IPS (AT-IPS) [34]. Note that we also compare with other debiasing methods such as knowledge distillation counterfactual learning [24], Causal Embedding [4], AutoDebias [5]. We discuss them in § 5.4 as they all require unbiased uniform data during training, which makes splitting datasets different.
Evaluation Protocol. For `Coat` and `Yahoo`, similar to previous works [34, 45], we use the original training set of the dataset as the training set and the unbiased uniform set as the test set. Note that feedbacks of the training set are MANR. We randomly selected 5% of the original training set as the validation set and adopted the unbiased evaluation method [37] to conduct the model section process on the validation set. For `Electronics`, we randomly sample 70% of user purchases as training data, 10% as validation, and the remaining 20% with the skewed sampling as held-out test data.
Overall Results. Table 1 shows the performance of LDR and baselines with NCF as the backbone. From this table, we have the following observations: (**i**) Overall, our LDR outperforms almost all baselines on all datasets, showing that our methods can effectively address the selection bias problem and achieve a better bias-variance trade-off. (**ii**) LDR can generally perform better than the baselines, which demonstrates the effectiveness and flexibility of LDR in facilitating various backbones. (**iii**) Our LDR significantly outperforms causal inference-based methods such as IPS, DR and AT-IPS. This can be explained by our iterative weighted learning: by using representation learning, our end-to-end solution can gradually improve weight estimation and reduce the bias of learning. (**iv**) Though both CVIB and ACL are not causal inference-based algorithms, our LDR outperforms them, which is because we can effectively estimate the importance weight and theoretically bound the variance.

5.3 (RQ2) Ablation and Sensitivity

Setup. To understand how different components affect the performance of LDR, we conduct an ablation study and hyper-parameter analysis. We follow the same experimental setting as RQ1. Specifically, we build the following variants: (**1**) LDR without the weight estimated by classifier c_ϕ (`LDR w/o C`); (**2**) LDR without representation adaptation by d_ψ (`LDR w/o D`). (3) `LDR-Static`: This is a static version of LDR that first pre-trains a representation with unweighted learning and representation adaptation. WE is conducted on the pre-trained representation, and then we conduct the unbiased WL.
Results. The results are reported in Table 2. From Table 2, we can find that both c_ϕ and d_ψ contribute to the performance gain, and

Table 1: Unbiased learning performance (%): NLL (↓), AUC (↑) and NDCG@10 (↑) using NCF of different methods.

Methods	Coat			Yahoo			Electronics		
	NLL ↓	**AUC ↑**	**NDCG@10 ↑**	**NLL ↓**	**AUC ↑**	**NDCG@10 ↑**	**NLL ↓**	**AUC ↑**	**NDCG@10 ↑**
Vanilla	53.24±0.56	74.29±0.07	69.12±0.13	55.27±1.05	66.85±0.12	75.20±0.26	65.63±0.13	58.11±0.19	37.82±0.36
IPS	52.08±2.39	75.09±1.32	70.13±0.87	53.39±3.56	67.22±1.53	76.21±1.98	63.11±2.08	59.23±1.58	38.97±2.33
DR	51.19±1.75	75.85±0.87	70.82±0.45	52.18±2.83	67.79±1.11	76.67±1.55	62.79±1.76	60.11±1.05	39.28±1.51
CVIB	49.55±0.77	**78.98**±0.22	71.89±0.19	47.36±1.01	69.03±0.52	77.35±0.75	60.37±0.31	62.04±0.71	41.25±0.75
ACL	50.43±1.51	76.11±0.47	71.22±0.23	49.33±1.81	68.43±1.27	76.82±0.96	61.65±0.98	61.97±0.83	39.81±1.11
AT-IPS	50.09±0.86	75.29±0.66	70.59±0.33	49.25±1.14	68.03±0.77	77.15±0.59	62.22±0.52	61.37±0.55	39.52±0.70
LDR	**48.81**±0.72	78.45±0.18	**72.82**±0.19	**45.25**±0.75	**70.22**±0.23	**78.94**±0.41	**59.01**±0.26	**63.33**±0.39	**42.38**±0.59

Table 2: Ablation study results with NCF as the backbone.

	Yahoo		Electronics	
	AUC	**NDCG@10**	**AUC**	**NDCG@10**
LDR	**70.22±0.23**	**78.94±0.41**	**63.33±0.39**	**42.38±0.59**
LDR w/o C	68.59±0.21	77.79±0.33	61.69±0.20	40.99±0.42
LDR w/o D	69.65±0.77	78.43±0.85	62.85±0.77	42.02±0.91
LDR-Static	68.32±0.35	77.37±0.41	61.77±0.28	41.58±0.52
Vanilla	66.85±0.12	75.20±0.26	58.11±0.19	37.82±0.36

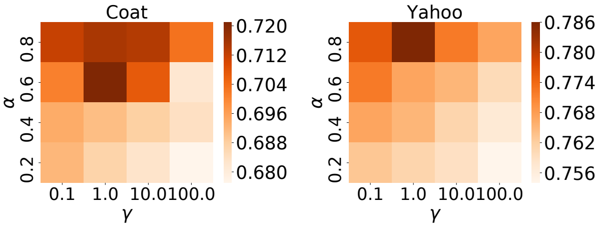


Figure 4: Parameter sensitivity analysis with NDCG@10.

their contributions are complementary to each other. Representation adaptation via D essentially boosts the performance and reduces the variance, while the de-biased weight component via C can further improve the performance. These results prove the effectiveness of employing both C and D in the proposed LDR. We also can observe that LDR can outperform LDR-Static, demonstrating that iteratively updating model parameters based on iterative reweighting of the training samples can improve performance.

To investigate hyper-parameter sensitivity, we vary the values of α and γ and report the NDCG@10 on Coat and Yahoo in Fig. 4. From the figure, we can find (**i**) Generally, with the increase of γ, the performance tends to first increase and then decrease. A too small γ would lead to a large variance and wrong weight estimation, while a large γ may dominate the whole loss of LDR. (**ii**) The performance is generally better and stable when α is between 0.6 and 0.8, which eases the parameter selection for LDR. (**iii**) We can balance the bias (weighted learning) and variance (representation adaptation) by varying α and γ, leading to better performance.

5.4 (RQ3) Debiasing with Partial Uniform Data

Setup. Recently, many works [4, 6, 24] show that incorporating uniform data in training can improve debiasing performance. Thus, we examine if LDR can also effectively utilize uniform data.

Baselines. We compare three representative baselines: Causal Embedding (CausE) [4], knowledge distillation counterfactual learning (KDCL) [24] (since there are several variants of KDCL, we choose the best result for comparison in each scenario) and AutoDebias [6]. We also compare IPS and DR since the uniform data can also improve their performance [24]. The way of using uniform data for our LDR is two folds: (i) adding uniform data into loss $\mathcal{L}_\omega$ (but no weight for uniform data); and (ii) utilizing the observed uniform data as $\mathcal{D}_q$ to conduct representation adaptation.

Table 3: Unbiased learning performance (%) using uniform data with NCF of different methods on unbiased test sets.

Methods	Yahoo		Electronics	
	AUC	**NDCG@10**	**AUC**	**NDCG@10**
Vanilla	76.19±0.11	76.28±0.18	60.28±0.13	40.98±0.17
IPS	77.22±1.21	76.55±0.99	62.36±2.57	41.55±2.89
DR	77.89±0.66	77.44±0.82	63.49±1.55	42.61±1.87
CausE	78.33±0.43	77.89±0.39	65.21±0.57	43.89±0.71
KDCL	78.99±0.35	78.57±0.19	66.80±0.49	45.21±0.52
AutoDebias	79.52±0.68	79.33±0.59	65.89±0.68	45.77±0.89
LDR	**80.58**±0.18	**80.31**±0.34	**68.53**±0.40	**46.88**±0.51

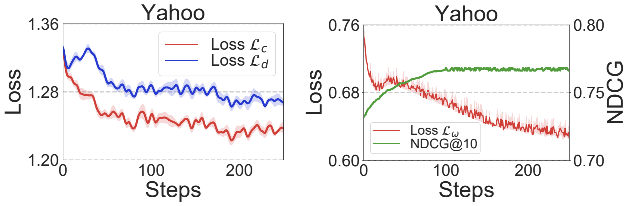


Figure 5: Curves of the weighted training losses and testing NDCG. The shaded area represents half a standard deviation.

Evaluation Protocol. Since we assume access to uniform data, the splitting is totally different from RQ1. We use all biased data as training data and randomly split the unbiased random subset into three subsets: 10% as the additional training data, 10% as the validation set, and the rest 80% as the test data.

Overall Results. Table 3 shows the results. It is shown that LDR can outperform all the other methods. These results validate the effectiveness of LDR for utilizing uniform data. Specifically, LDR can still outperform IPS and DR, which once again shows that narrowing the gap between WE and WL can improve unbiased learning performance. Our LDR outperforms KDCL and AutoDebias, which validates that our LDR with theoretical unbiasedness and variance guarantee can provide improvements in the recommendation quality on biased datasets.

5.5 (RQ4) Model Analysis

Setup. We take a deeper examination on the proposed LDR to understand how it works. We follow the same setting as RQ1.

Convergence. The instability [16] in min-max training is well known. Thus, we investigate the training process of LDR. Fig. 5 shows the curves of the training losses and the testing NDCG. Results on other datasets also share a similar tendency. From the figure,

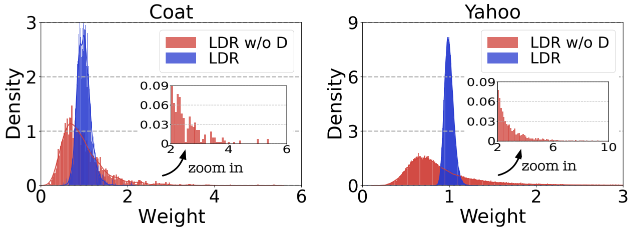


Figure 6: The densities of weight distributions.

we can find: (1) LDR is training-stable and can consistently improve performance as training steps increase. (2) Generally, LDR has a small variance in adversarial and classification losses throughout the whole training. (3) The results show that NDCG can converge within a few hundred steps, which is efficient.

Bounded Weight and Variance. We investigate whether the representation adaptation can bound the weight and variance. Following the same settings as § 5.3, we consider the ablation `LDR w/o D`. The learned weights are given in Fig. 6. We find that the weights learned by LDR generally have lower variance than that learned by `LDR w/o D`. This shows that our proposed representation adaptation is effective for bounding the variance. It also reduce the largest weight, i.e., w_m in § 4.3, which leads to a smaller bias in Eq. (9).

The Bias and Variance Reduction. To better understand how LDR contributes to learning an unbiased and low variance ranking model, we take a deeper examination on the weight distribution grouped by the interacted frequency of items (item popularity). From Fig. 7, we can observe: (1) As the item interacted frequency increases, the mean weight generally decreases. Thus, LDR can successfully identify unpopular/popular items and automatically up-/down-weight them. This confirms that LDR can improve weight learning and thus reduce selection bias. (2) Generally, the variance of the weights for the items is small, although the variance of the weights of the popular items is larger than unpopular. Meanwhile, as shown in Table 2 and Fig. 6, our LDR not only achieves better performance but also attains lower estimation variance than other ablation variants such as `LDR w/o D` and `LDR w/o C`.

6 CONCLUSION

In this paper, we study the problem of handling the selection bias from MNAR feedback. We propose Learning De-biased Representations (LDR), a general framework to address this problem. Specifically, LDR embeds learned representations into the dynamic and iterative procedure to yield a more reliable weight estimate and leverages the representation adaptation to reduce the variance. Theoretical analysis proves its unbiasedness and desired statistical properties. Empirical experiments on several datasets show that LDR achieves better performance under selection bias.

ACKNOWLEDGMENTS

This work is partially supported by the National Science Foundation (NSF) under grants number IIS-1909702 and IIS1955851, and Army Research Office (ARO) under grant number W911NF-21-1-0198.

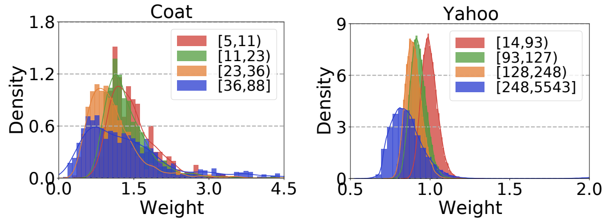


Figure 7: The weight distributions with the item popularity.

A PROOFS

A.1 Proofs of Theorem 4.1

Proof. By the definition of probability density functions (PDFs) and the fundamental theorem of calculus, we have:

$$w(\mathbf{x}_u, \mathbf{x}_i) = \frac{q(\mathbf{x}_u, \mathbf{x}_i)}{p(\mathbf{x}_u, \mathbf{x}_i)} = \frac{q(\mathbf{x}_u, \mathbf{x}_i)|[\frac{\partial f_\theta^{-1}(\mathrm{y})}{\partial \mathrm{y}}]_{\mathrm{y}=f_\theta(\mathbf{x}_u,\mathbf{x}_i)}|}{p(\mathbf{x}_u, \mathbf{x}_i)|[\frac{\partial f_\theta^{-1}(\mathrm{y})}{\partial \mathrm{y}}]_{\mathrm{y}=f_\theta(\mathbf{x}_u,\mathbf{x}_i)}|}$$
$$= \frac{q(f_\theta(\mathbf{x}_u, \mathbf{x}_i))}{p(f_\theta(\mathbf{x}_u, \mathbf{x}_i))} = \frac{q(\mathbf{z}_{ui})}{p(\mathbf{z}_{ui})} = w(\mathbf{z}_{ui}), \tag{17}$$

which completes the proof. □

A.2 Proofs of Theorem 4.2

Proof. For brevity, we define that $WE(\mathcal{D}_p, \mathcal{D}_q) = \widehat{w}(\mathcal{D}_p, \mathcal{D}_q)$ and $WE(f_\theta(\mathcal{D}_p), f_\theta(\mathcal{D}_q)) = \widehat{w}(f_\theta(\mathcal{D}_p), f_\theta(\mathcal{D}_q))$. We also define that $\widehat{w}(\mathcal{D}_p, \mathcal{D}_q)(\mathbf{x}_u, \mathbf{x}_i) = \widehat{w}(\mathbf{x}_u, \mathbf{x}_i)$ is the empirical weight of the sample pair $w(\mathbf{x}_u, \mathbf{x}_i)$, and $\widehat{w}(f_\theta(\mathcal{D}_p), f_\theta(\mathcal{D}_q))(f_\theta(\mathbf{x}_u, \mathbf{x}_i)) = \widehat{w}(f_\theta(\mathbf{x}_u, \mathbf{x}_i)) = \widehat{w}(\mathbf{z}_{ui})$ is the empirical weight of the transformed $\mathbf{z}_{ui} = f_\theta(\mathbf{x}_u, \mathbf{x}_i)$. Given the definitions, we can obtain:

$$\mathbb{E}_{p(f_\theta(\mathbf{x}_u,\mathbf{x}_i)),q(f_\theta(\mathbf{x}_u,\mathbf{x}_i))}[\widehat{w}(f_\theta(\mathbf{x}_u, \mathbf{x}_i))]$$
$$= \int p(f_\theta(\mathbf{x}_u, \mathbf{x}_i))q(f_\theta(\mathbf{x}_u, \mathbf{x}_i))[\widehat{w}(f_\theta(\mathbf{x}_u, \mathbf{x}_i))]df_\theta(\mathbf{x}_u, \mathbf{x}_i)$$
$$= \int p(\mathbf{x}_u, \mathbf{x}_i)q(\mathbf{x}_u, \mathbf{x}_i)[\widehat{w}(f_\theta(\mathbf{x}_u, \mathbf{x}_i))]d(\mathbf{x}_u, \mathbf{x}_i)$$
$$= \mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i),q(\mathbf{x}_u,\mathbf{x}_i)}[\widehat{w}(f_\theta(\mathbf{x}_u, \mathbf{x}_i))]$$
$$= \frac{q(f_\theta(\mathbf{x}_u, \mathbf{x}_i))}{p(f_\theta(\mathbf{x}_u, \mathbf{x}_i))} = \frac{q(\mathbf{x}_u, \mathbf{x}_i)}{p(\mathbf{x}_u, \mathbf{x}_i)} = \mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i),q(\mathbf{x}_u,\mathbf{x}_i)}[\widehat{w}(\mathbf{x}_u, \mathbf{x}_i)], \tag{18}$$

where we use Theorem 4.1 in the last equation. This completes the proof that $\widehat{w}(f_\theta(\mathcal{D}_p), f_\theta(\mathcal{D}_q))(f_\theta(\mathbf{x}_u, \mathbf{x}_i)) = \widehat{w}(f_\theta(\mathbf{x}_u, \mathbf{x}_i)) = \hat{w}(\mathbf{z}_{ui})$ is an unbiased empirical estimation of ideal $w(\mathbf{x}_u, \mathbf{x}_i)$. □

A.3 Proofs of Theorem 4.3

Proof. With the estimated weight $\widehat{w}(\mathbf{z}_{ui})$, the bias of weighted learning $\mathcal{L}_w(\omega, \theta)$ can be derived as follows:

$$|\mathcal{L}_{ideal}(\omega) - \widehat{\mathcal{L}}_w(\omega, \theta)| = |\mathbb{E}_p[(w(\mathbf{z}_{ui}) - \widehat{w}(\mathbf{z}_{ui}))\ell(h_\omega(\mathbf{z}_{ui}), y)]| \leq$$
$$\sqrt{\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}\left[(w(\mathbf{z}_{ui}) - \widehat{w}(\mathbf{z}_{ui}))^2\right]\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(\ell(h_\omega(\mathbf{z}_{ui}), y))^2]} \tag{19}$$
$$\leq \frac{1}{2}(\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(w(\mathbf{x}_{ui}) - \widehat{w}(\mathbf{z}_{ui}))^2] + \mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(\ell(h_\omega(\mathbf{x}_u, \mathbf{x}_i), y))^2])$$

where the second line holds due to the Cachy-Schwarz inequality and the third line holds due to the inequality of arithmetic and geometric means (AM–GM inequality). As mentioned in the main body of the paper, the optimal weight can be estimated via a classifier:

$$\widehat{w}(\mathbf{z}_{ui}) = k\frac{\widehat{r}(d = 0|\mathbf{z}_{ui})}{\widehat{r}(d = 1|\mathbf{z}_{ui})} = k(\frac{1}{\widehat{r}(d = 1|\mathbf{z}_{ui})} - 1), \tag{20}$$

where $k = \frac{r(d=1)}{r(d=0)}$ is a constant and we set it as 1 in what follows without loss of generality. Since $\widehat{w}(\mathbf{z}_{ui}) \leq w_m$, we have $\frac{1}{w_m+1} \leq \widehat{r}(d = 1 \mid \mathbf{x}) \leq 1$. Given this, we have:

$$
\begin{aligned}
&\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(w(\mathbf{z}_{ui}) - \widehat{w}(\mathbf{z}_{ui}))^2] \\
&= \mathrm{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(\frac{r(d=1|\mathbf{z}_{ui}) - \widehat{r}(d=1|\mathbf{z}_{ui})}{r(d=1|\mathbf{z}_{ui})\widehat{r}(d=1|\mathbf{z}_{ui})})^2] \\
&\leq (w_m+1)^4 \mathrm{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}\left[(r(d=1|\mathbf{z}_{ui}) - \widehat{r}(d=1|\mathbf{z}_{ui}))^2\right].
\end{aligned}
\tag{21}
$$

Plugging this into Eq. (19) completes the proof. □

A.4 Proofs of Theorem 4.4

Proof. We first rewrite the estimation variance as follows (Note that we define $w(\mathbf{z}_{ui})\mathcal{L}(\omega,\theta)) \triangleq \mathcal{L}_w(\omega,\theta) \triangleq \mathcal{L}_w$):

$$
\begin{aligned}
&\mathrm{Var}[\mathcal{L}_w] = \mathrm{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[(\ell_w)^2] - (\mathrm{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\ell_w])^2 \qquad (22)\\
&= \mathrm{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[w(\mathbf{z}_{ui})^2(\ell)^2] - (\mathrm{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\ell_w])^2 \\
&= \int p(\mathbf{x}_u,\mathbf{x}_i,y)[(\frac{q(\mathbf{z}_{ui})}{p(\mathbf{z}_{ui})})^2(\ell)^2] - (\mathrm{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\ell_w])^2 d(\mathbf{x}_u\mathbf{x}_i)dy \\
&= \int q(\mathbf{z}_{ui},y)^{\frac{1}{\alpha}} \frac{q(\mathbf{z}_{ui},y)}{p(\mathbf{z}_{ui},y)} q(\mathbf{z}_{ui},y)^{\frac{\alpha-1}{\alpha}} (\ell)^2 - (\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w])^2 d\mathbf{z}_{ui}dy,
\end{aligned}
$$

where the last line holds since we change variables and $p(y|\mathbf{z}_{ui}) = q(y|\mathbf{z}_{ui})$. By using Hölder’s inequality, we can bound the it as:

$$
\begin{aligned}
&\int q(\mathbf{z}_{ui},y)^{\frac{1}{\alpha}} \frac{q(\mathbf{z}_{ui},y)}{p(\mathbf{z}_{ui},y)} q(\mathbf{z}_{ui},y)^{\frac{\alpha-1}{\alpha}} (\ell)^2 - (\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w])^2 d\mathbf{z}_{ui}dy \leq \\
&(\int q(\mathbf{z}_{ui},y)(\frac{q(\mathbf{z}_{ui},y)}{p(\mathbf{z}_{ui},y)})^{\alpha})^{\frac{1}{\alpha}} (\int q(\mathbf{z}_{ui},y)\ell^{\frac{2\alpha}{\alpha-1}})^{\frac{\alpha-1}{\alpha}} - (\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w])^2 = \\
&d_{\alpha+1}(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))(\int q(\mathbf{z}_{ui},y)\ell\ell^{\frac{\alpha+1}{\alpha-1}})^{\frac{\alpha-1}{\alpha}} - (\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w])^2 \quad (23)\\
&\leq d_{\alpha+1}(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))(\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w)]^{\frac{\alpha-1}{\alpha}} (\int \ell)^{1+\frac{1}{\alpha}} - (\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w])^2 \\
&\leq d_{\alpha+1}(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))\left(\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w]\right)^{\frac{\alpha-1}{\alpha}} - \left(\mathrm{E}_{p(\mathbf{z}_{ui},y)}[\ell_w]\right)^2 = \\
&d_{\alpha+1}(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))(\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\mathcal{L}_w])^{\frac{\alpha-1}{\alpha}} - (\mathbb{E}_{p(\mathbf{x}_u,\mathbf{x}_i,y)}[\mathcal{L}_w])^2,
\end{aligned}
$$

$\forall \alpha > 0$, where we use Hölder’s inequality in the first inequality and the second inequality holds since $(\int \ell)^{1+\frac{1}{\alpha}} > 1$. Plugging this inequality into Eq. (22) completes the proof. □

A.5 Proofs of Theorem 4.5

Proof. By using Theorem 4.4 with $\alpha = 1$, we have:

$$
\mathrm{Var}[\mathcal{L}_w] \leq d_2(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui})). \tag{24}
$$

Since we have $\widehat{w}(\mathbf{z}_{ui}) = w(\mathbf{z}_{ui})$, we only use $w(\mathbf{z}_{ui})$ for the sake of simplicity. By using Bernstein inequality, we have

$$
\begin{aligned}
&P(|\mathcal{L}_{ideal}(\omega) - \widehat{\mathcal{L}}_w(\omega,\theta)| > \epsilon) = P(|\mathcal{L}_w(\omega,\theta) - \widehat{\mathcal{L}}_w(\omega,\theta)| > \epsilon) \\
&P(|\mathcal{L}_w(\omega,\theta) - \frac{1}{N}\sum_{(\mathbf{x}_u,\mathbf{x}_i,y)\in\mathcal{D}_p} w(\mathbf{z}_{ui})\,\ell(h_\omega(f_\theta(\mathbf{x}_u,\mathbf{x}_i)),y)\,| > \epsilon) \\
&P(|\mathcal{L}_w(\omega,\theta) - \frac{1}{N}\sum_{(\mathbf{x}_u,\mathbf{x}_i,y)\in\mathcal{D}_p} w(\mathbf{z}_{ui})\,\ell(h_\omega(f_\theta(\mathbf{x}_u,\mathbf{x}_i)),y)\,| > \epsilon) \\
&\leq \exp\left(\frac{-N\epsilon^2/2}{\mathrm{Var}[\mathcal{L}_w] + \epsilon w_m l_m/3}\right).
\end{aligned}
\tag{25}
$$

Setting δ to $\exp\left(\frac{-N\epsilon^2/2}{\mathrm{Var}[\mathcal{L}_w]+\epsilon w_m l_m/3}\right)$ and solving ϵ yields:

$$
P(|\mathcal{L}_{ideal}(\omega) - \widehat{\mathcal{L}}_w(\omega,\theta)| < B) \leq 1 - \sigma, \tag{26}
$$

Algorithm 1: The Training Algorithm for LDR

```
Input: Factual observation dataset
  $\mathcal{D}_p \triangleq \{x_u^{(n)}, x_i^{(n)}, y^{(n)}\}_{n=1}^N$, counterfactual dataset
  $\mathcal{D}_q \triangleq \{x_u^{(m)}, x_i^{(m)}\}_{m=1}^M$ sampled from uniform distribution
  $q(\mathbf{x}_u,\mathbf{x}_i) = q(\mathbf{x}_u)q(\mathbf{x}_i)$. Hyper-parameters $\alpha$ and $\gamma$.
  Learning rate $\eta_\omega$, $\eta_\theta$, $\eta_\phi$, and $\eta_\psi$.
Initialize: $h_\omega$, $f_\theta$, $c_\phi$ and $d_\psi$.
for $t = 1, \cdots$, num iterations do
    Sample mini-batches of $(x_u, x_i, y) \in \mathcal{D}_p$ and
  $(x_u, x_i) \in \mathcal{D}_q$
    $\omega_t \leftarrow \omega_{t-1} - \eta_\omega \nabla_\omega \mathcal{L}_w(\omega,\theta)$ ▶ Unbiased Weighted
  loss
    $\phi_t \leftarrow \phi_{t-1} - \eta_\phi \alpha \nabla_\phi \mathcal{L}_c(\phi)$ ▶ Classification loss
    $\theta_t \leftarrow \theta_{t-1} - \eta_\theta \nabla_\theta \gamma(\mathcal{L}_d(\theta,\psi_{t-1})) + \mathcal{L}_w(\omega_t,\theta))$ ▶
  Min-step
    $\psi_t \leftarrow \psi_{t-1} + \eta_\psi \nabla_\psi \gamma \mathcal{L}_d(\theta_t,\psi))$ ▶ Max-step
Return $\omega$, $\phi$, $\theta$, and $\psi$.                ▶ Optimized parameters
```

where $B = \frac{w_m l_m \log 1/\delta}{3N} + \sqrt{\frac{l_m^2 w_m^2 \log^2 1/\delta}{9N^2} + \frac{2l_m^2 \mathrm{Var}(\mathcal{L}_w)\log 1/\delta}{N}}$. Plugging Eq. (24) into B and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we have:

$$
B \leq \frac{2l_m w_m \log 1/\sigma}{3N} + l_m\sqrt{\frac{2d_2(q(\mathbf{z}_{ui})\|p(\mathbf{z}_{ui}))\log 1/\sigma}{N}}. \tag{27}
$$

Plugging this into Eq. (26) completes the proof. □

B DATASET DETAILS

Yahoo!R3[1]: It contains five-star ratings. The biased training set contains 311,704 MNAR five-star ratings of 1,000 songs from 15,400 users, and the unbiased test set contains ratings collected by asking 5,400 users to rate ten randomly selected songs. Coat[2]: It contains five-star ratings from 290 users and 300 items. The training set contains 6,960 MNAR ratings collected via user self-selections, and the test set is collected by asking users to rate 16 uniformly selected items. Amazon-Electronics[3] dataset: It contains 7,824,482 five-star rating from 33,602 users and 16,448 items. To create the unbiased test set from Electronics, we conduct a skewed splitting strategy following [4, 34], which exposes each user as uniformly as possible to each item in the testing set. Specifically, we sample a test set from the original dataset, then re-sample data from the test set based on the inverse of the probabilities as:

$$
p_i = \frac{\sum_{u\in\mathcal{U}} O_{u,i}}{\max_{i\in I}\sum_{u\in\mathcal{U}} O_{u,i}}, \tag{28}
$$

where $O_{u,i}$ indicts if the feedback is observed or not: $\left[O_{u,i} = 1\right] \Leftrightarrow \left[y_{u,i} \text{ is observed }\right]$. This skewed splitting strategy creates a synthetic test set that each item has a uniform observed probability.

[1] https://webscope.sandbox.yahoo.com/

[2] https://www.cs.cornell.edu/~schnabts/mnar/

[3] https://nijianmo.github.io/amazon/index.html

REFERENCES

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. 2014. Domain-adversarial neural networks. In *arXiv*.

[2] Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. 2021. A contrastive learning approach for training variational autoencoder priors. In *Conference on Neural Information Processing Systems*.

[3] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2007. Discriminative learning for differing training and test distributions. In *International Conference on Machine Learning*.

[4] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *ACM Recommender Systems conference*.

[5] 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 *ACM SIGIR Conference on Research and Development in Information Retrieval*.

[6] 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. In *arXiv*.

[7] Zhengyu Chen, Sibo Gai, and Donglin Wang. 2019. Deep Tensor Factorization for Multi-Criteria Recommender Systems. In *International Conference on Big Data (Big Data)*. 1046–1051.

[8] Zhengyu Chen and Donglin Wang. 2021. Multi-Initialization Meta-Learning with Domain Adaptation. In *The International Conference on Acoustics, Speech, & Signal Processing*. 1390–1394.

[9] Zhengyu Chen, Ziqing Xu, and Donglin Wang. 2021. Deep transfer tensor decomposition with orthogonal constraint for recommender systems. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*. 3.

[10] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. 2010. Learning Bounds for Importance Weighting. In *Conference on Neural Information Processing Systems*.

[11] Imre Csiszár. 1967. Information-type measures of difference of probability distributions and indirect observation. In *Studia Sci. Math. Hungar.*

[12] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. In *International Conference on Machine Learning*.

[13] Harrison Edwards and Amos J. Storkey. 2016. Censoring Representations with an Adversary. In *International Conference on Learning Representations*.

[14] Benjamin Eysenbach, Shreyas Chaudhari, Swapnil Asawa, Sergey Levine, and Ruslan Salakhutdinov. 2020. Off-Dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers. In *International Conference on Learning Representations*.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. In *The Journal of Machine Learning Research*.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Conference on Neural Information Processing System*.

[17] Negar Hassanpour and Russell Greiner. 2019. CounterFactual Regression with Importance Sampling Weights. In *International Joint Conference on Artificial Intelligence*.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *The World Wide Web Conference*.

[19] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. In *Neural networks*.

[20] Fredrik Johansson, Uri Shalit, and David Sontag. 2016. Learning representations for counterfactual inference. In *International Conference on Machine Learning*.

[21] Fredrik D Johansson, Nathan Kallus, Uri Shalit, and David Sontag. 2018. Learning weighted representations for generalization across designs. In *arXiv*.

[22] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. In *Computer*.

[23] Zinan Lin, Dugang Liu, Weike Pan, and Zhong Ming. 2021. Transfer Learning in Collaborative Recommendation for Bias Reduction. In *ACM Recommender Systems Conference*.

[24] 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 *ACM SIGIR Conference on Research and Development in Information Retrieval*.

[25] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. Mitigating Confounding Bias in Recommendation via Information Bottleneck. In *ACM Recommender Systems Conference*.

[26] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

[28] Benjamin M Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative filtering and the missing at random assumption. In *UAI*.

[29] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Conference on Empirical Methods in Natural Language Processing*.

[30] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *Conference on Neural Information Processing System*.

[31] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *ACM Recommender Systems Conference*.

[32] Alfréd Rényi et al. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*.

[33] Noveen Sachdeva, Yi Su, and Thorsten Joachims. 2020. Off-policy bandits with deficient support. In *KDD*.

[34] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-Not-At-Random Explicit Feedback. In *ACM SIGIR Conference on Research and Development in Information Retrieval*.

[35] Yuta Saito. 2020. Doubly robust estimator for ranking metrics with post-click conversions. In *ACM Recommender Systems Conference*.

[36] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *ACM International Conference on Web Search and Data Mining*.

[37] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning*.

[38] Uri Shalit, Fredrik D Johansson, and David Sontag. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*.

[39] Xinwei Sun, Botong Wu, Xiangyu Zheng, Chang Liu, Wei Chen, Tao Qin, and Tie-Yan Liu. 2021. Recovering Latent Causal Factor for Generalization to Distributional Shifts. In *Conference on Neural Information Processing System*.

[40] Adith Swaminathan and Thorsten Joachims. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. In *The Journal of Machine Learning Research*.

[41] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *Conference on Neural Information Processing System*.

[42] Takeshi Teshima, Issei Sato, and Masashi Sugiyama. 2020. Few-shot domain adaptation by causal mechanism transfer. In *International Conference on Machine Learning*.

[43] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning*.

[44] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2021. Combating Selection Biases in Recommender Systems with a Few Unbiased Ratings. In *ACM International Conference on Web Search and Data Mining*.

[45] Zifeng Wang, Xi Chen, Rui Wen, Shao-Lun Huang, Ercan Kuruoglu, and Yefeng Zheng. 2020. Information Theoretic Counterfactual Learning from Missing-Not-At-Random Feedback. In *Conference on Neural Information Processing System*.

[46] Teng Xiao and Donglin Wang. 2021. A General Offline Reinforcement Learning Framework for Interactive Recommendation. In *AAAI Conference on Artificial Intelligence*. 4512–4520.

[47] Teng Xiao and Suhang Wang. 2022. Towards Off-Policy Learning for Ranking Policies with Logged Feedback. In *AAAI Conference on Artificial Intelligence*. 8700–8707.

[48] Teng Xiao and Suhang Wang. 2022. Towards unbiased and robust causal ranking for recommender systems. In *ACM Conference on Web Search and Data Mining*. 1158–1167.

[49] Yuan Xie, Boyi Liu, Qiang Liu, Zhaoran Wang, Yuan Zhou, and Jian Peng. 2018. Off-Policy Evaluation and Learning from Logged Bandit Feedback: Error Reduction via Surrogate Policy. In *International Conference on Learning Representations*.

[50] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Adversarial Counterfactual Learning and Evaluation for Recommender System. In *Conference on Neural Information Processing System*.

[51] Da Xu, Yuting Ye, and Chuanwei Ruan. 2020. Understanding the role of importance weighting for deep learning. In *International Conference on Learning Representations*.

[52] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. 2018. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*.

[53] 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 *The Web Conference*.

[54] Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. 2020. Unbiased Implicit Recommendation and Propensity Estimation via Combinational Joint Learning. In *ACM Recommender Systems Conference*.