Counterfactual Implicit Feedback Modeling

Chuan Zhou[1,2] **Lina Yao**[3,4] **Haoxuan Li**[5,2,*] **Mingming Gong**[1,2,*]
[1]The University of Melbourne [2]Mohamed bin Zayed University of Artificial Intelligence
[3]The University of New South Wales [4]CSIRO's Data61 [5]Peking University
chuan.zhou@student.unimelb.edu.au, lina.yao@data61.csiro.au,
hxli@stu.pku.edu.cn, mingming.gong@unimelb.edu.au

Abstract

In recommendation systems, implicit feedback data can be automatically recorded and is more common than explicit feedback data. However, implicit feedback poses two challenges for relevance prediction, namely (a) **positive-unlabeled** (PU): negative feedback does not necessarily imply low relevance and (b) **missing not at random** (MNAR): items that are popular or frequently recommended tend to receive more clicks than other items, even if the user does not have a significant interest in them. Existing methods either overlook the MNAR issue or fail to account for the inherent mechanism of the PU issue. As a result, they may lead to inaccurate relevance predictions or inflated biases and variances. In this paper, we formulate the implicit feedback problem as a counterfactual estimation problem with missing treatment variables. Prediction of the relevance in implicit feedback is equivalent to answering the counterfactual question that "*whether a user would click a specific item if exposed to it?*". To solve the counterfactual question, we propose the Counterfactual Implicit Feedback (Counter-IF) prediction approach that divides the user-item pairs into four disjoint groups, namely definitely positive (DP), highly exposed (HE), highly unexposed (HU), and unknown (UN) groups. Specifically, Counter-IF first performs missing treatment imputation with different confidence levels from raw implicit feedback, then estimates the counterfactual outcomes via causal representation learning that combines pointwise loss and pairwise loss based on the user-item pairs stratification. Theoretically the generalization bound of the learned model is derived. Extensive experiments are conducted on publicly available datasets to demonstrate the effectiveness of our approach. The code is available at `https://github.com/zhouchuanCN/NeurIPS25-Counter-IF`.

1 Introduction

Recommender systems are technical tools that analyze historical user behavioral data to predict their preferences and actively provide personalized recommendations [1, 2, 3], which have been widely applied to various fields, such as e-commerce, streaming video, and social media [4, 5, 6, 7]. There are two types of feedback mechanisms in user behavioral data, including explicit feedback [8, 9, 10], which refers to the explicit signals that users directly express their preferences, such as product ratings. Implicit feedback [11], on the other hand, comes from the indirect interactions between users and the system, including unstructured behavioral trajectories such as the length of time spent on the page, clicks, and so on. Implicit feedback has a great advantage over explicit feedback in terms of data acquisition [12, 5], as most user behaviors are implicit [13], which can be collected continuously on a large scale and can reflect the potential user demand more promptly.

In implicit feedback recommendation, the goal is to infer the user's true relevance or preference for an item from their behavior data, i.e., to determine whether the user is likely to be interested in an item.

[*]Haoxuan Li and Mingming Gong are the corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

However, this process faces two key challenges. The first one is the **positive-unlabeld** (PU) learning problem [14, 15]. The system only observes a binary signal of whether the user clicks the item, but the clicking behavior does not fully reflect user preferences. Specifically, the items not clicked on may be attributed to user disinterest, or they may not even be exposed to the user. The second challenge is that the feedback data is **missing not at random** (MNAR) [16, 17]. For example, items frequently recommended tend to attract more clicks, even if the user does not have a substantial interest in them. In recent years, studies have begun to notice that MNAR in implicit feedback data can introduce bias into predictions, thus damaging the performance of the recommendation system [18, 19, 20].

In the development of implicit feedback recommendation, early approaches such as weighted matrix factorization (WMF) [21], exposure perception matrix factorization (ExpoMF) [8] and Bayesian personalized ranking (BPR) [13] are based on the key assumption that the observed feedbacks directly reflect the true preferences, whereas unobserved feedbacks are uniformly treated as negative samples or given low weights. However, early methods assume that the missing mechanism is missing completely at random (MCAR), ignoring the MNAR nature of implicit data, leading to biased relevance estimation [16]. Recent studies attempt to introduce causal techniques into implicit feedback modeling, e.g., through counterfactual bias correction using propensity score [22, 23]. However, these causal learning methods still have some drawbacks. The first one is that propensity models are prone to be overly confident, generating extremely inaccurate propensity score estimation [24, 25, 26]. And like any propensity-based method, the bias and variance of the estimator can be extremely large with small propensity [27, 28], thus affecting the effectiveness of relevance prediction. The second one is that existing causal recommendation methods for implicit feedback data usually use EM algorithms to estimate the exposure propensity, but the intrinsic positive-unlabeled mechanism of implicit feedback is neglected, which may hinder the model from accurately estimating the propensity.

In this paper we propose that implicit feedback recommendation can be addressed by answering the counterfactual question: **whether a user would click a specific item if exposed to it?** To tackle this problem, we formulate it as a counterfactual outcome prediction problem with missing treatment. Let the feedback label be the outcome, and the exposure be the treatment. To answer the counterfactual question, we actually need to estimate the potential outcome corresponding to the exposed treatment group, with treatment of only the exposed group observed. Although there has been work dealing with the MNAR problem with explicit feedback data [27, 29, 30, 31], and remarkably much more work in the area of statistics estimating causal effects [32, 33], none of them could be applied to the counterfactual problem formulated in this paper. That is because all previous causal methods require the treatment to be observed for every sample in the dataset. However, this counterfactual problem brings unique challenges with missing treatment.

To overcome these challenges, we propose the **Counter**factual **I**mplicit **F**eedback (Counter-IF) method for estimating counterfactual outcomes in the presence of missing treatment variables in implicit feedback scenarios. To uncover more information from negative samples based on the positive-unlabeled nature, the Counter-IF consists of stratifying user-item pairs in implicit feedback. Specifically, using the estimated confidence, we impute the exposure only for samples with high confidence. Apart from the **definitely positive** (DP) samples group, we classify the negative samples into three groups based on the different reasons leading to unclick: those with **high probability of being exposed** (HE) tend to have low relevance and those with **high probability of being unexposed** (HU) tend to have higher relevance than other **unknown** (UN) negative samples. Counter-IF also includes a causal representation learning framework that combines pointwise and pairwise losses based on the imputed treatments, leading to accurate counterfactual outcomes estimation.

The main contributions can be summarized as follows:

- We are the first to formalize the relevance prediction problem under implicit feedback scenarios as a counterfactual outcome estimation problem with missing treatments.
- We propose a sample stratification algorithm in Counter-IF for implicit feedback using a treatment variable imputation method with confidence, reflecting different mechanisms of negative sample generation.
- We propose a causal representation learning framework in Counter-IF to answer the formalized counterfactual questions. We theoretically derive the generalization bound of our causal learning model.
- We conduct extensive experiments on publicly available real-world datasets, demonstrating the proposed Counter-IF significantly outperforms state-of-the-art methods.

2 Problem Setup of Implicit Feedback

Let $u \in \mathcal{U}$ denote a user, $i \in \mathcal{I}$ be an item and $\mathcal{D} = \mathcal{U} \times \mathcal{I}$ be the set of all user-item pairs. The complete set consists of $|\mathcal{U}| \times |\mathcal{I}|$ user-item pairs. The recommender system observes only the implicit feedback $Y_{u,i} \in \{0, 1\}$, which passively captures whether the user u clicks the item i. The feature of (u, i) is denoted as $X_{u,i}$. We divide all user-item pairs according to $Y_{u,i}$, i.e., $\mathcal{D}_1 = \{(u, i) \mid (u, i) \in \mathcal{D}, Y_{u,i} = 1\}$ and $\mathcal{D}_0 = \{(u, i) \mid (u, i) \in \mathcal{D}, Y_{u,i} = 0\}$. Due to the sparsity of positive samples in implicit feedback data, we have $|\mathcal{D}_1| \ll |\mathcal{D}_0|$. A positive feedback $Y_{u,i} = 1$ directly implies that u and i are relevant. However, we are not certain whether a negative feedback $Y_{u,i} = 0$ indicates the item is irrelevant to the user since it depends on whether u is exposed to i.

We consider the exposure $O_{u,i}$ as a treatment, with $O_{u,i} = 1$ meaning that the user u is exposed to the item i and vice versa. Note that we cannot observe $O_{u,i}$ for all the user-item pairs in $\mathcal{D}$. Instead, we can merely infer that for those $(u, i) \in \mathcal{D}_1$, the user u must be exposed to item i, thus $O_{u,i} = 1$. However, for those $(u, i) \in \mathcal{D}_0$, $O_{u,i}$ is unknown. To more clearly formulate the implicit feedback problem, we introduce the true relevance score $S_{u,i} \in \{0, 1\}$, with $S_{u,i} = 1$ indicating that u and i are relevant and vice versa. Following previous work [16], we assume in this paper that:

$$Y_{u,i} = O_{u,i} \cdot S_{u,i}. \tag{1}$$

Given $Y_{u,i} = 1, O_{u,i} = 1$ for $(u, i) \in \mathcal{D}_1$ and $Y_{u,i} = 0$ for $(u, i) \in \mathcal{D}_0$, we would like to predict $S_{u,i}$ for all $(u, i) \in \mathcal{D}$. Note that Equation (1) implies $Y_{u,i} = 1 \iff O_{u,i} = 1$ and $S_{u,i} = 1$.

3 Methodology

3.1 Causal Formulation of Implicit Feedback

The question we aim to answer in implicit feedback recommendation is "whether a user would click a specific item if exposed to it", a typical counterfactual question. Therefore, it is logical to express it in causal terminology. In this paper, we employ the potential outcome framework. In our causal formulation, we let $O_{u,i}$ be the treatment, and $Y_{u,i}(1)$ be the potential outcome of feedback if forcing $O_{u,i} = 1$, and $Y_{u,i}(0)$ be the potential outcome of feedback if we force $O_{u,i} = 0$. We require the consistency assumption, which means that if u is exposed to i, the observed outcome of feedback is the potential outcome we aim to estimate, i.e., $Y_{u,i} = O_{u,i}Y_{u,i}(1) + (1 - O_{u,i})Y_{u,i}(0)$. We also assume that the stable unit treatment value assumption (SUTVA) holds, i.e., there should be only one form of exposure between u and i, and there is no interference between user-item pairs. In addition, we assume that the unconfoundedness assumption holds, i.e., $Y_{u,i}(1) \perp\!\!\!\perp O_{u,i} | X_{u,i}$. In the recommendation scenario, this assumption implies that the intrinsic relevance between u and i is independent of whether to expose u to i. Then on the one hand, by the consistency assumption we have $O_{u,i} = 0 \Rightarrow Y_{u,i} = Y_{u,i}(0)$. Meanwhile, by Equation (1), we have $O_{u,i} = 0 \Rightarrow Y_{u,i} = 0$. As a consequence, $Y_{u,i}(0) = 0$ for $(u, i) \in \mathcal{D}$. On the other hand, we have $O_{u,i} = 1 \Rightarrow Y_{u,i} = Y_{u,i}(1)$ and $O_{u,i} = 1 \Rightarrow Y_{u,i} = S_{u,i}$. Combining the two equations derived under the condition $O_{u,i} = 1$, we have $Y_{u,i}(1) = S_{u,i}$ for $(u, i) \in \mathcal{D}$.

Table 1: The user-item pairs are divided into four subgroups from a principal stratification perspective, named *Definitely Positive* group, *Highly Exposed* group, *Highly Unexposed* group, and *Unknown* group. The red font indicates that the value is imputed.

Group	O	R	Y	$Y(1)$
Definitely Positive (DP)	1	1	1	1
Highly Exposed (HE)	1	1	0	0
Highly Unexposed (HU)	0	1	0	?
Unknown (UN)	?	0	0	?

Therefore, we transform the estimation of $S_{u,i}$ into the estimation of $Y_{u,i}(1)$. In other words, we formulate the implicit feedback problem as **missing treatment counterfactual problem**, where the treatment assignments to the samples with negative feedback are unknown, and we aim to estimate the potential outcomes $Y_{u,i}(1)$. With the consistency assumption, we can infer that the potential outcome $Y_{u,i}(1) = 1$ for $(u, i) \in \mathcal{D}_1$, since for these samples $Y_{u,i} = 1$, we can infer $O_{u,i} = 1$, then $Y_{u,i}(1) = Y_{u,i} = 1$. However, for $(u, i) \in \mathcal{D}_0$, we are not sure whether $O_{u,i} = 1$ or $O_{u,i} = 0$, so we cannot directly infer the value of $Y_{u,i}(1)$ for this group of user-item pairs.

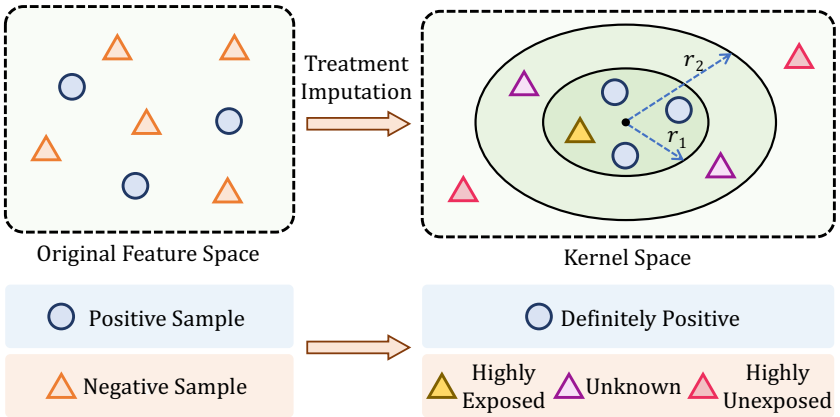


Figure 1: Missing treatment imputation with different confidence level from implicit feedback.

3.2 Stratification of User-Item Pairs

For the purpose of counterfactual estimation, it is almost impossible to solely rely on the outcome variables and a small portion of the treatment variables. So we, therefore, wanted to dig deeper into the intrinsic mechanisms of the implicit feedback data to get more information about exposure. Unlike previous methods that impute treatments using a consistent scheme for all samples in $\mathcal{D}$, based on the diversity of meanings of negative samples in implicit feedback, we consider imputation only for those items the system is highly confident to recommend to a user. Specifically, we define a binary indicator $R_{u,i} \in \{0, 1\}$, which reflects the confidence of the recommender system in assigning the treatment. $R_{u,i} = 1$ means that the system is with high confidence, and $R_{u,i} = 0$ means that the system is not confident about treatment assignment. We denote the estimated confidence as $\hat{R}_{u,i}$ for our treatment imputation model. For those with high confidence, we impute the treatment as the predicted $\hat{O}_{u,i}$. Then based on $(\hat{O}_{u,i}, \hat{R}_{u,i}, Y_{u,i})$, we divide all user-item pairs into four strata:

- Definitely Positive (DP) Group $\mathcal{D}_{DP}$: This group consists of user-item pairs with $Y_{u,i}(1) = 1$, the imputation method should be definitely confident about the corresponding treatment $O_{u,i} = 1$, meaning u has definitely been exposed to i. We have $\mathcal{D}_{DP} = \mathcal{D}_1$.
- For user-item pairs in $\mathcal{D}_0$, we further classify them into three categories, according to the estimated confidence indicator $\hat{R}_{u,i}$ and imputed treatment $\hat{O}_{u,i}$:
  - Highly Exposed (HE) Group: The imputation method has high confidence that the user is exposed to the item ($\hat{R}_{u,i} = 1$ and $\hat{O}_{u,i} = 1$), thus impute the treatment as 1, meaning the user tend to be exposed to the item ($O_{u,i} = 1$). However, the observed $Y_{u,i} = 0$, which means that $Y_{u,i}(1)$ is likely to be 0.
  - Highly Unexposed (HU) Group: The imputation method has high confidence that the user is unexposed to the item ($R_{u,i} = 1$ and $\hat{O}_{u,i} = 0$), thus impute the treatment as 0. $Y_{u,i}(1)$ is more likely to be 1 than other negative samples.
  - Unknown (UN) Group: The imputation method has low confidence to assign treatment ($R_{u,i} = 0$), so we do not impute for the sample ($O_{u,i} =?$), keeping its multiple interpretation nature.

As Table 1 shows, we can infer the value of $Y_{u,i}(1)$ for the DP and HE groups, but not for the HU and UN groups. For the DP group $Y_{u,i}(1) = Y_{u,i} = 1$. For the HE group, given the imputation is accurate, we have $Y_{u,i}(1) = Y_{u,i} = 0$.

3.3 Treatment Imputation with Confidence

Intuitively, we expect the DP group, consisting of user-item pairs known to be exposed, and the HE group, which includes user-item pairs for whom the system has high confidence that they are exposed, to be close to each other in a specific feature space. On the other hand, the HU group, composed of user-item pairs for whom the system is confident that are unexposed, should be far

away from the exposed groups (HE & DP). The remaining users, whose exposure status remains uncertain, form the Unknown group (UN), positioned between the positive and negative groups. To divide users into the four subgroups, we propose a novel exposure estimation approach to impute the treatments for solving the counterfactual problem, as shown in Figure 1. Inspired by support vector domain description [34], we propose a novel treatment imputation method that utilizes the distance requirement between the four groups to assign samples to groups. Specifically, with feature $x_{u,i} \in \mathcal{X}$ as the input, we enclose a proportion of α samples of DP within a hypersphere, where $\alpha \in (0, 1)$ is a hyperparameter. Then we assign the negative samples within the hypersphere into the HE group. Then the farthest β proportion of the negative samples is assigned to the HU group, where $\beta \in (0, 1)$ is another hyperparameter. The remaining negative samples go to the UN group.

Our treatment assignment method aims to encapsulate α proportion of positive samples within a hypersphere defined by its center a and radius r. The optimization problem can be formulated as:

$$\begin{aligned} &\min_{r,a,\epsilon} F(r,a) = r^2 + C \sum_{(u,i)\in\mathcal{D}_{DP}} \epsilon_{u,i}, \\ &\text{s.t.} \quad \|x_{u,i} - a\|^2 \leq r^2 + \epsilon_{u,i}, \epsilon_{u,i} \geq 0, \forall (u,i) \in \mathcal{D}_{DP}, \end{aligned}$$

where r is the radius of the hypersphere, a is its center, and $\epsilon_{u,i}$ are slack variables that allow certain user-item samples to lie outside the hypersphere.

To solve this optimization problem, we incorporate the Lagrange multipliers $\lambda_{u,i} \geq 0$ and $\gamma_{u,i} \geq 0$ for each constraint:

$$L = r^2 + C \sum_{(u,i)\in\mathcal{D}_{DP}} \left\{ \epsilon_{u,i} - \lambda_{u,i} \left(r^2 + \epsilon_{u,i} - (x_{u,i}^2 - 2a x_{u,i} + a^2) \right) - \gamma_{u,i}\epsilon_{u,i} \right\}.$$

Then, the dual optimization problem can be represented as:

$$\begin{aligned} &\max_{\lambda} \sum_{(u,i)\in\mathcal{D}_{DP}} \lambda_{u,i}(x_{u,i} \cdot x_{u,i}) - \sum_{(u,i),(u',i')\in\mathcal{D}_{DP}} \lambda_{u,i}\lambda_{u',i'}(x_{u,i} \cdot x_{u',i'}), \\ &\text{s.t.} \quad 0 \leq \lambda_{u,i} \leq C. \end{aligned}$$

To handle non-linearity in the data, we apply a kernel function $K(x_{u,i}, x_{u',i'})$, which implicitly maps the input data into a higher-dimensional feature space. A commonly used kernel is the Gaussian radial basis function (RBF) kernel, defined as:

$$K(x_{u,i}, x_{u',i'}) = \exp(-q\|x_{u,i} - x_{u',i'}\|),$$

where q controls the width of the kernel and thus the smoothness of the decision boundary.

The squared distance of a sample x from the center of the hypersphere is computed as:

$$d^2(x) = K(x,x) - 2 \sum_{(u,i)\in\mathcal{D}_{DP}} \lambda_{u',i'} K(x_{u,i}, x) + \sum_{(u,i),(u',i')\in\mathcal{D}_{DP}} \lambda_{u,i}\lambda_{u',i'} K(x_{u,i}, x_{u',i'}).$$

And the squared radius of the hypersphere is $d^2(x_v)$. where x_v is a support vector.

Then, we find the smallest α and the largest β proportion of $\{d^2(x_{u,i}) \mid (u,i) \in \mathcal{D}_0\}$, and denote the threshold as r_1 and r_2 respectively to classify $\mathcal{D}_0$ into the following groups:

$$\begin{aligned} (u,i) \in \mathcal{D}_{HE} \quad &\text{if} \quad d^2(x_{u,i}) \leq r_1, \\ (u,i) \in \mathcal{D}_{UN} \quad &\text{if} \quad r_1 < d^2(x_{u,i}) \leq r_2, \\ (u,i) \in \mathcal{D}_{HU} \quad &\text{if} \quad d^2(x_{u,i}) > r_2. \end{aligned}$$

3.4 Counterfactual Representations

After obtaining a stratification for the samples, we propose a causal representation learning method to predict the relevance between users and items across $\mathcal{D}$. When $O_{u,i} = 1$, indicating that the relevance scores for samples in HE and DP groups are observed, we can use pointwise loss to train the model based on the outcomes $Y(1)$. This ensures that the model effectively fits the potential outcome. In contrast, when the samples are not assigned the treatment group (i.e., $O_{u,i} = 0$, corresponding to the UN and HU groups), we treat these unobserved interactions as counterfactual data and employ

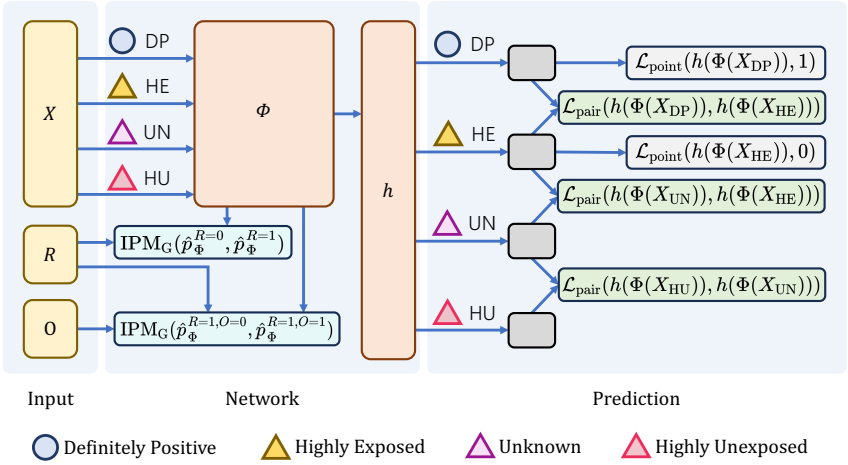


Figure 2: Representation learning framework through a counterfactual lens: exploring four strata of implicit feedback.

pairwise loss to model the relative rankings. This approach enhances the capability of our model to capture user-item relevance across all samples.

The user-item feature $x_{u,i}$ is transformed into the representation $\Phi(x_{u,i})$. The prediction model h is then applied to the representation $\Phi(x_{u,i})$ to output the predicted interaction outcome $h(\Phi(x_{u,i}))$.

For DP and HE samples, we apply a pointwise cross-entropy loss. This ensures that the model accurately predicts the relevance based on the observed data. Specifically, for the DP samples, where $Y(1) = 1$, we have:

$$\mathcal{L}_{\text{point}}(h(\Phi(X_{DP})), 1) = -\frac{1}{|\mathcal{D}_{DP}|} \sum_{(u,i)\in\mathcal{D}_{DP}} \log h(\Phi(x_{u,i})).$$

While for the HE samples, where $Y(1) = 0$, the loss is defined as:

$$\mathcal{L}_{\text{point}}(h(\Phi(X_{HP})), 0) = -\frac{1}{|\mathcal{D}_{HE}|} \sum_{(u,i)\in\mathcal{D}_{HE}} \log(1 - h(\Phi(x_{u,i}))).$$

Therefore, the whole point loss is defined below:

$$\mathcal{L}_1 = \mathcal{L}_{\text{point}}(h(\Phi(X_{DP})), 1) + \mathcal{L}_{\text{point}}(h(\Phi(X_{HP})), 0).$$

For the unobserved interactions (i.e., $O_{u,i} = 0$, corresponding to X_{UN} and X_{HU}), we use the pairwise loss to model the ranking of the interaction outcomes. Specifically, the following pairwise loss is applied to optimize the relative ranking between strata:

$$\mathcal{L}_{\text{pair}}(h(\Phi(X_+)), h(\Phi(X_-))) = \frac{1}{|\mathcal{D}_+| \cdot |\mathcal{D}_-|} \sum_{(u,i)\in\mathcal{D}_+,(j,k)\in\mathcal{D}_-} \log(\sigma(h(\Phi(x_{u,i}))) - h(\Phi(x_{j,k})),$$

where $\mathcal{D}_+$ and $\mathcal{D}_-$ are the sets of positive samples and negative samples. The pairwise ranking loss encourages the model to assign higher scores to positive samples over negative samples.

Specifically, we observe that the interaction probability for the HE group is significantly greater than that for the UN group, while the interaction probability for the HU group is markedly lower than that for the UN group. We apply the interaction probabilities among various user-item groups to establish positive and negative sample pairs and obtain the counterfactual loss:

$$\mathcal{L}_2 = \mathcal{L}_{\text{pair}}(h(\Phi(X_{DP})), h(\Phi(X_{HE})) + \mathcal{L}_{\text{pair}}(h(\Phi(X_{UN})), h(\Phi(X_{HE})) + \mathcal{L}_{\text{pair}}(h(\Phi(X_{HU})), h(\Phi(X_{UN})).$$

To mitigate distributional shifts between different strata of samples, we introduce Integral Probability Metric (IPM) regularization. IPM ensures that the representations of samples with confidence ($R = 1$) and samples that are not sure ($R = 0$) are aligned in the representation space. Furthermore, we also

introduce regularization for $R = 1, O = 0$ and $R = 1, O = 1$ samples to ensure consistency. The IPM regularization terms are defined as:

$$\mathcal{L}_{\text{IPM}} = \text{IPM}_G(p_\Phi^{R=0}, p_\Phi^{R=1}) + \text{IPM}_G(p_\Phi^{R=1,O=0}, p_\Phi^{R=1,O=1}).$$

By introducing IPM, we aim to ensure that the model learns representations that generalize well across different sample distributions. The IPM is defined as:

$$\text{IPM}_G(p, q) := \sup_{g \in G} \left| \int_S g(s)(p(s) - q(s))\, ds \right|,$$

where p and q are two probability distributions, and G is a class of functions for which we seek to optimize the difference in expectations.

The total loss for the model incorporates both pointwise and pairwise losses, addressing the observed and unobserved interactions across different strata. The final relevance prediction model is trained by minimizing the following loss:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{point}} \mathcal{L}_1 + \lambda_{\text{pair}} \mathcal{L}_2 + \mathcal{L}_{\text{IPM}}.$$

3.5 Theoretical Analysis

We theoretically derive the generalization bound under our framework and show that minimizing the proposed pointwise loss $\mathcal{L}_1$, pairwise loss $\mathcal{L}_2$, and IPM loss $\mathcal{L}_{\text{IPM}}$ can effectively control the bound. First, we introduce the following assumption to ensure the existence of the inverse representation:

Assumption 3.1 (Inverse Representation and Function Class [35])**.** The representation $\Phi : \mathcal{X} \to \mathcal{A}$ is a one-to-one function, with inverse Ψ. Let G be a family of functions $g : \mathcal{A} \to \mathcal{Y}$. Assume there exists a constant $B_\Phi > 0$, such that $\frac{1}{B_\Phi} \cdot (h \circ \Phi \circ \Psi(a) - Y(1))^2 \in \text{G}$.

Based on Assumption 3.1, we then derive the following generalization bound:

Theorem 3.2 (Generalization Bound)**.** *Under Assumption 3.1, the deviation between the estimated relevance $h(\Phi(x))$ and expected relevance $m_1(x) = \mathbb{E}[Y(1) \mid X = x]$ averaging on all user-item pairs has the upper bound:*

$$\begin{aligned}
&\mathbb{E}_x[(h(\Phi(x)) - m_1(x))^2] \leq \underbrace{\mathbb{E}_{x|r,o}[(h(\Phi(x)) - Y(1))^2 \mid R = 1, O = 1]}_{\textit{factual loss of the DP and HE groups}} \\
&+ \mathbb{P}(O = 0 \mid R = 1) \cdot B_\Phi \cdot \underbrace{IPM_G(p_\Phi^{R=1,O=0}, p_\Phi^{R=1,O=1})}_{\textit{distribution shift on O given R=1}} + \underbrace{\mathbb{P}(R = 0)}_{\textit{UN group}} \cdot B_\Phi \cdot \underbrace{IPM_G(p_\Phi^{R=0}, p_\Phi^{R=1})}_{\textit{distribution shift on R}} \\
&- \underbrace{\mathbb{E}[(Y(1) - m_1(x))^2]}_{\textit{variance of potential outcome}}.
\end{aligned}$$

In the above bound, the first term is the factual loss based on the true value of $Y(1)$ of the DP and HE groups, which can be controlled by minimizing the $\mathcal{L}_{\text{point}}$ and $\mathcal{L}_{\text{pair}}$. The second and third terms are the IPM distance measuring the distribution shift on $O = 1$ and $O = 0$ given $R = 1$ group and distribution shift on $R = 1$ and $R = 0$ weighted by the proportion of HU group given $R = 1$ and proportion of UN group $\mathbb{P}(R = 0)$, respectively. These IPM distance terms can also be effectively controlled by minimizing the proposed $\mathcal{L}_{\text{IPM}}$. Intuitively, if $\mathbb{P}(R = 0) = 0$, there is no need to control the distribution shift on R. The last term measures the minimal variance of potential outcomes, which is independent of the model selection. See Appendix A for the detailed proof.

4 Experiments

4.1 Experimental Setup

Datasets. To evaluate the performance of unbiased recommendations, we utilize two real-world datasets: **Coat** and **Yahoo! R3**. Each dataset includes both biased training data and an unbiased test set. The **Coat** dataset contains 6,960 biased ratings and 4,640 unbiased ratings provided by 290 users for 300 items, where each user rates 16 randomly selected items. The **Yahoo! R3** dataset includes

Table 2: Ranking performance on Yahoo and Coat. We bold the best results and underline the best baseline. The results with * indicate p < 0.05 using the pairwise t-test with the best competitor.

	Methods	K=3			K=5			K=8		
		NDCG@K	Recall@K	MAP@K	NDCG@K	Recall@K	MAP@K	NDCG@K	Recall@K	MAP@K
Yahoo	ExpoMF	0.524 ± 0.008	0.581 ± 0.012	0.461 ± 0.008	0.588 ± 0.007	0.736 ± 0.010	0.509 ± 0.007	0.652 ± 0.005	0.912 ± 0.003	0.548 ± 0.006
	WMF	0.538 ± 0.005	0.596 ± 0.006	0.470 ± 0.004	0.600 ± 0.005	0.755 ± 0.011	0.529 ± 0.004	0.663 ± 0.004	0.918 ± 0.004	0.561 ± 0.004
	Rel-MF	0.534 ± 0.008	0.599 ± 0.009	0.465 ± 0.008	0.593 ± 0.007	0.749 ± 0.010	0.523 ± 0.007	0.653 ± 0.004	0.918 ± 0.003	0.555 ± 0.005
	Rel-MF-du	0.540 ± 0.008	0.596 ± 0.008	0.478 ± 0.009	0.611 ± 0.007	0.756 ± 0.010	0.530 ± 0.009	0.668 ± 0.007	0.915 ± 0.009	0.557 ± 0.008
	BPR	0.517 ± 0.003	0.574 ± 0.005	0.455 ± 0.003	0.581 ± 0.006	0.732 ± 0.011	0.502 ± 0.005	0.654 ± 0.004	0.905 ± 0.005	0.542 ± 0.004
	UBPR	0.532 ± 0.005	0.592 ± 0.005	0.470 ± 0.005	0.596 ± 0.002	0.746 ± 0.007	0.517 ± 0.003	0.657 ± 0.002	0.913 ± 0.003	0.555 ± 0.004
	UBPR-nclip	0.536 ± 0.008	0.597 ± 0.008	0.474 ± 0.009	0.597 ± 0.006	0.746 ± 0.010	0.522 ± 0.009	0.659 ± 0.007	0.914 ± 0.007	0.557 ± 0.008
	UPL	0.546 ± 0.005	0.603 ± 0.011	0.483 ± 0.004	0.610 ± 0.004	0.759 ± 0.007	0.532 ± 0.005	0.668 ± 0.004	0.922 ± 0.007	0.568 ± 0.004
	RecRec	0.545 ± 0.007	0.602 ± 0.004	0.484 ± 0.008	0.607 ± 0.004	0.757 ± 0.004	0.533 ± 0.006	0.669 ± 0.004	0.923 ± 0.008	0.570 ± 0.006
	Ours	**$0.562^* \pm 0.007$**	**$0.624^* \pm 0.009$**	**$0.499^* \pm 0.007$**	**$0.625^* \pm 0.005$**	**$0.776^* \pm 0.008$**	**$0.547^* \pm 0.006$**	**$0.681^* \pm 0.004$**	**0.930 ± 0.004**	**$0.582^* \pm 0.005$**
Coat	ExpoMF	0.324 ± 0.007	0.340 ± 0.010	0.256 ± 0.007	0.372 ± 0.006	0.459 ± 0.012	0.287 ± 0.007	0.428 ± 0.008	0.601 ± 0.008	0.315 ± 0.008
	WMF	0.333 ± 0.016	0.322 ± 0.019	0.264 ± 0.015	0.369 ± 0.016	0.412 ± 0.020	0.294 ± 0.015	0.426 ± 0.016	0.610 ± 0.025	0.325 ± 0.015
	RelMF	0.338 ± 0.013	0.344 ± 0.018	0.267 ± 0.010	0.385 ± 0.005	0.462 ± 0.012	0.304 ± 0.006	0.433 ± 0.008	0.614 ± 0.022	0.338 ± 0.006
	RelMF-du	0.340 ± 0.016	0.332 ± 0.023	0.275 ± 0.013	0.374 ± 0.010	0.420 ± 0.014	0.307 ± 0.009	0.458 ± 0.015	0.640 ± 0.021	0.353 ± 0.012
	BPR	0.324 ± 0.011	0.325 ± 0.018	0.265 ± 0.010	0.370 ± 0.010	0.433 ± 0.017	0.290 ± 0.008	0.445 ± 0.009	0.640 ± 0.017	0.335 ± 0.007
	UBPR	0.343 ± 0.012	0.342 ± 0.018	0.269 ± 0.012	0.384 ± 0.009	0.451 ± 0.017	0.306 ± 0.009	0.449 ± 0.009	0.642 ± 0.023	0.339 ± 0.008
	UBPR-nclip	0.335 ± 0.007	0.345 ± 0.013	0.261 ± 0.006	0.368 ± 0.011	0.430 ± 0.022	0.290 ± 0.009	0.445 ± 0.007	0.640 ± 0.012	0.335 ± 0.006
	UPL-BPR	0.345 ± 0.009	0.343 ± 0.014	0.273 ± 0.009	0.377 ± 0.009	0.427 ± 0.025	0.302 ± 0.007	0.438 ± 0.009	0.615 ± 0.014	0.340 ± 0.008
	RecRec	0.360 ± 0.008	0.365 ± 0.014	0.284 ± 0.005	0.392 ± 0.009	0.452 ± 0.019	0.314 ± 0.006	0.454 ± 0.007	0.629 ± 0.018	0.354 ± 0.004
	Ours	**0.368 ± 0.011**	**$0.382^* \pm 0.012$**	**$0.296^* \pm 0.009$**	**$0.414^* \pm 0.012$**	**$0.478^* \pm 0.014$**	**$0.332^* \pm 0.009$**	**$0.473^* \pm 0.012$**	**0.660 ± 0.021**	**$0.369^* \pm 0.009$**

Table 3: Ablation study on the Yahoo and Coat datasets.

	Methods	K=3			K=5			K=8		
		NDCG@K	Recall@K	MAP@K	NDCG@K	Recall@K	MAP@K	NDCG@K	Recall@K	MAP@K
Yahoo	w/o Wass w/o Pair	0.546 ± 0.008	0.609 ± 0.009	0.483 ± 0.009	0.611 ± 0.007	0.765 ± 0.008	0.531 ± 0.009	0.668 ± 0.007	0.923 ± 0.009	0.566 ± 0.004
	w/o Wass w/o Point	0.552 ± 0.007	0.615 ± 0.010	0.488 ± 0.006	0.616 ± 0.008	0.770 ± 0.008	0.536 ± 0.007	0.673 ± 0.006	0.927 ± 0.005	0.572 ± 0.005
	w/o Pair	0.552 ± 0.009	0.615 ± 0.010	0.488 ± 0.008	0.616 ± 0.006	0.770 ± 0.009	0.536 ± 0.007	0.673 ± 0.007	0.927 ± 0.007	0.572 ± 0.006
	w/o Point	0.554 ± 0.005	0.614 ± 0.007	0.491 ± 0.006	0.619 ± 0.007	0.774 ± 0.004	0.540 ± 0.007	0.675 ± 0.008	0.927 ± 0.006	0.575 ± 0.007
	w/o Wass	0.558 ± 0.008	0.623 ± 0.010	0.494 ± 0.009	0.620 ± 0.009	0.771 ± 0.009	0.540 ± 0.008	0.676 ± 0.006	0.927 ± 0.005	0.576 ± 0.008
	All	**0.562 ± 0.007**	**0.624 ± 0.009**	**0.499 ± 0.007**	**0.625 ± 0.005**	**0.776 ± 0.008**	**0.547 ± 0.006**	**0.681 ± 0.004**	**0.930 ± 0.004**	**0.582 ± 0.005**
Coat	w/o Wass w/o Pair	0.362 ± 0.010	0.376 ± 0.010	0.280 ± 0.009	0.398 ± 0.011	0.477 ± 0.012	0.314 ± 0.008	0.450 ± 0.010	0.618 ± 0.019	0.347 ± 0.007
	w/o Wass w/o Point	0.363 ± 0.009	0.378 ± 0.010	0.285 ± 0.009	0.399 ± 0.010	0.477 ± 0.013	0.319 ± 0.009	0.450 ± 0.011	0.609 ± 0.020	0.352 ± 0.009
	w/o Pair	0.366 ± 0.011	0.366 ± 0.012	0.290 ± 0.008	0.412 ± 0.010	**0.485 ± 0.015**	0.327 ± 0.011	0.472 ± 0.013	0.646 ± 0.022	0.368 ± 0.009
	w/o Point	0.361 ± 0.009	0.363 ± 0.010	0.289 ± 0.009	0.407 ± 0.010	0.484 ± 0.013	0.325 ± 0.009	0.455 ± 0.011	0.614 ± 0.021	0.356 ± 0.009
	w/o Wass	0.364 ± 0.010	0.377 ± 0.011	0.281 ± 0.010	0.402 ± 0.012	0.478 ± 0.014	0.317 ± 0.010	0.468 ± 0.012	0.656 ± 0.020	0.357 ± 0.008
	All	**0.368 ± 0.011**	**0.382 ± 0.012**	**0.296 ± 0.009**	**0.414 ± 0.012**	0.478 ± 0.014	**0.332 ± 0.009**	**0.473 ± 0.012**	**0.660 ± 0.021**	**0.369 ± 0.009**

311,704 biased ratings and 54,000 unbiased ratings from 15,400 users interacting with 1,000 items. The positive samples are sparse, which is consistent with the real-world situation. We employed the preprocessing steps following previous studies [23, 36], which can be seen in the Appendix B.

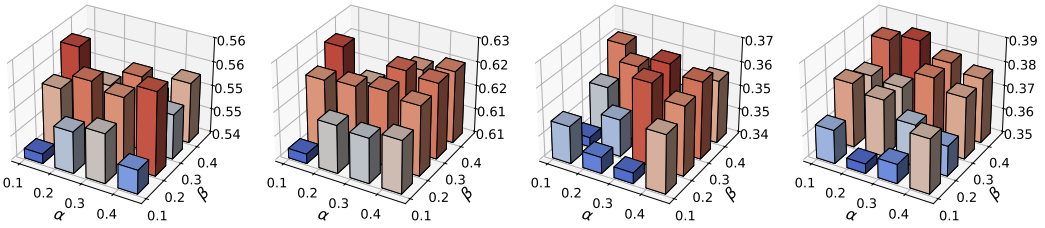
Evaluation Metrics and Details. We use three common metrics to evaluate implicit recommendation systems: NDCG@k (Normalized Discounted Cumulative Gain), Recall@k, and MAP@k (Mean Average Precision). DCG@k evaluates the ranking quality by giving more weight to relevant items appearing earlier in the list. Recall@k measures how many relevant items are retrieved within the top k recommendations. MAP@k calculates the mean precision across users, considering both relevance and order. Results are presented for $k = 3$, $k = 5$, and $k = 8$.

Hyperparamter Tuning. For each dataset, the data was divided into training and test sets. A portion of 10% from the training set was randomly selected to serve as the validation set for hyperparameter tuning. Several key parameters were adjusted during this phase. The latent factor dimensions, representing user-item interactions, were explored within the range of 100 to 300, while the L2 regularization term was fine-tuned between $[10^{-7}, 10^{-3}]$ for all models, and the λ_{point} as well as λ_{pair} are tuned in $\{0.01, 0.1, 1, 10, 100\}$.

Baselines. To achieve a comprehensive comparison, we consider the following methods as baselines: **WMF** [1], **ExpoMF** [8], **Rel-MF** [16], **Rel-MF-du** [16], **BPR** [13], **UBPR** [23], **UBPR-nclip** [23], **UPL** [37], and **RecRec** [36].

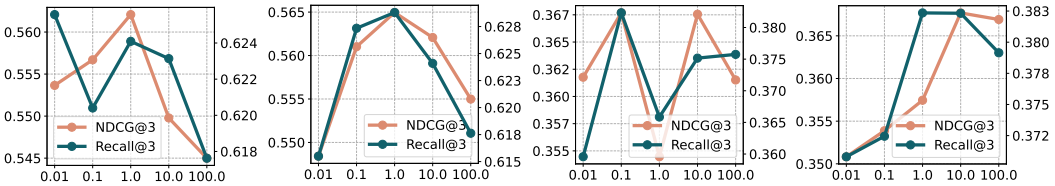
4.2 Performance Comparison

We evaluate the performance of our proposed method against several baseline approaches on multiple datasets, as shown in Table 2. The results highlight several key findings. Traditional methods like BPR and WMF show moderate performance but struggle with the PU and MNAR challenges in implicit feedback. BPR misclassifies potential positives as negatives by assuming unobserved interactions are negative, while WMF treats unobserved interactions as low-weight positives, failing to fully address false negatives. Methods like ExpoMF and Rel-MF improve by modeling exposure and item popularity, reducing bias. ExpoMF incorporates exposure variables but remains limited by traditional matrix factorization, while Rel-MF leverages item popularity to estimate propensities, though it still



(a) NDCG@3 on Yahoo (b) Recall@3 on Yahoo (c) NDCG@3 on Coat (d) Recall@3 on Coat

Figure 3: Sensitivity analysis on the proportion of the HE group α and the proportion of the HU group β for samples in $\mathcal{D}_0$.



(a) λ_{pair} on Yahoo (b) λ_{point} on Yahoo (c) λ_{pair} on Coat (d) λ_{point} on Coat

Figure 4: The sensitivity of balancing weight between pointwise and pairwise losses.

struggles with unobserved items. UBPR and UBPR-nclip extend BPR to address PU and MNAR problems, with UBPR-nclip further reducing bias through a non-clipping estimator. UPL simplifies pairwise learning, performing well in high-variance scenarios. Finally, ReCRec shows strong results but is computationally intensive, limiting scalability. Despite improvements over simpler models, these baselines still fail to fully address the challenges of implicit feedback. Our method outperforms all baselines by explicitly modeling exposure and preference to address PU and MNAR issues. By combining the proposed treatment imputation and balanced representation learning, our method leads to more accurate predictions.

4.3 Ablation Study

We conduct ablation studies to validate the effectiveness of our model by examining the impact of the Wasserstein distance (Wass), pairwise loss (Pair), and pointwise loss (Point). Table 3 shows performance metrics (NDCG@K, Recall@K, MAP@K) on Yahoo and Coat datasets. Removing any component degrades performance, as each addresses positive-unlabeled and MNAR challenges in implicit feedback. Removing both Wass and Pair (w/o Wass w/o Pair) yields the lowest performance, as the model cannot stratify negative samples or learn robust representations. Similarly, removing Wass and Point (w/o Wass w/o Point) compromises the causal learning framework. Removing only Pair (w/o Pair) or Point (w/o Point) also reduces performance, though less severely. Both losses contribute uniquely: Pair improves ranking, while Point predicts click likelihood under exposure. Removing only Wass (w/o Wass) degrades performance, as the model loses the ability to stratify negative samples by exposure probability, crucial for distinguishing unclicked items due to low relevance versus lack of exposure. In summary, the proposed model with all components (Wass, Pair, Point) achieves the best performance, highlighting the importance of each in addressing implicit feedback challenges and estimating counterfactual outcomes with missing treatments.

4.4 Sensitivity Analysis

Threshold and proportion. Figure 3 shows the sensitivity of the model's performance on the proportion of the HE group α and the proportion of the HU group β for samples in $\mathcal{D}_0$. With very low or very high α, performance tends to degrade because the model potentially excludes relevant samples or includes irrelevant samples. Similarly, an extreme β degrades the performance.

Coefficients of the loss function. As shown in Figure 4, we investigate the impact of the coefficients λ_{pair} and λ_{point} for the pairwise and pointwise losses in the proposed model. We evaluate the effects of varying these coefficients on NDCG@K, Recall@K, and MAP@K across the Yahoo! R3 and Coat datasets. The results show that setting λ_{pair} and λ_{point} within a moderate range (e.g., 0.1, 1, or 10)

leads to significant improvements in ranking accuracy and click prediction. Excessively large values (e.g., 100) overemphasize either ranking or click prediction, degrading performance. Similarly, very small values (e.g., 0.01) fail to leverage both losses effectively, resulting in poor performance. The optimal performance is achieved when these coefficients are set to 1 or 10, striking a balance between ranking accuracy and click likelihood prediction.

5 Related Work

5.1 Implicit Feedback

Early work solving the positive-unlabeled problems in implicit recommender systems includes weighted matrix factorization (WMF), and some approaches like MF [1] downweight negative samples uniformly, some reweight samples via user activities [38], and others use item popularity to adjust weights [39]. Exposure models, such as ExpoMF [8] models exposure probabilities based on item popularity and text topics, while other methods incorporate social or community data [12, 40]. Most methods rely on pointwise loss, but pairwise loss (e.g., Bayesian personalized ranking, BPR [13]) is better suited for ranking tasks by learning relative preferences. However, these methods do not consider the implicit feedback in MNAR, thus may get biased relevance prediction.

Recently, there are some propensity-based methods proposed to address positive-unlabeled and MNAR issues. Rel-MF [22] leverages item popularity for propensity estimation and solves MNAR problem in implicit feedback, while joint learning approaches [41, 42] infer both propensity and recommendation models. Methods using small unbiased datasets employ embedding alignment [6], knowledge distillation [43], or meta-learning [44] to learn exposure/propensity models. For pairwise loss, UBPR [23] proposed a debiased loss. However, the propensity models are prone to be overly confident, generating extremely inaccurate propensity score estimation [25, 26, 45]. And the bias and variance of the estimator can be extremely large with small propensity [27, 28].

5.2 Causal Recommendation

Causal recommendations refer to applying the causal frameworks [46, 47, 48] to predict the conversion rates [49] and provide personalized recommendations [50, 51]. Compared with previous recommendation methods, the causal recommendations can address various biases, including the confounding bias [52, 53, 54], popularity bias [55], selection bias [18, 56, 57], and exposure bias [58], which previous methods fail to address because they highly rely on the associations. Most of the methods are based on inverse-propensity or doubly robust weighting [59, 60, 61]. Some studies propose to use the same training and inference space with entire-space multi-task learning approaches [62, 63, 28, 64]. They jointly train the parameters to achieve better recommendation performance. More recent works focus on counterfactual learning, which predicts the counterfactual outcomes for the user-item pairs [65, 66, 67]. However, most of them cannot achieve individual-level counterfactual predictions. [68] extends the above methods to perform individual counterfactual predictions. However, the above methods are mainly applied in explicit feedback recommendations. It is impossible to directly apply the above methods to the implicit feedback because it is challenging to determine whether an observed negative user-item pair is irrelevant in implicit feedback recommender systems.

6 Conclusion

In this paper, we focus on the problem of inferring the true relevance or preference of a user in the implicit feedback recommendation scenarios. Specifically, to the best of our knowledge, we are the first paper to formalize the relevance prediction problem as a counterfactual outcome estimation problem with missing treatments, which provides a novel approach to tackle this problem. Correspondingly, we propose a sample stratification method, which uses a treatment variable imputation method with feature similarity-based confidence, reflecting different mechanisms of negative sample generation. In addition, we propose a balanced representation-based causal learning framework to answer the formalized counterfactual questions and theoretically derive the generalization bound of our causal learning model, showing that minimizing the proposed loss functions can effectively control the bound. Extensive experiments on public benchmark datasets show the effectiveness of our method. One potential limitation is that the method requires pre-specification of the distance thresholds. Changing the framework to an end-to-end manner may further help improve performance.

Acknowledgments and Disclosure of Funding

The authors thank the anonymous reviewers for their valuable comments. HL was supported by the National Natural Science Foundation of China (623B2002). MG was supported by ARC DP240102088 and WIS-MBZUAI 142571.

References

[1] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[2] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *RecSys*, 2016.

[3] Chunyuan Zheng, Hang Pan, Yang Zhang, and Haoxuan Li. Adaptive structure learning with partial parameter sharing for post-click conversion rate prediction. In *SIGIR*, 2025.

[4] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *IUI*, 2010.

[5] Dietmar Jannach, Lukas Lerche, and Markus Zanker. Recommending based on implicit feedback. In *Social information access: systems and technologies*, pages 510–569. Springer, 2018.

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

[7] Beibei Li, Beihong Jin, Jiageng Song, Yisong Yu, Yiyuan Zheng, and Wei Zhou. Improving micro-video recommendation via contrastive multiple interests. In *SIGIR*, 2022.

[8] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. Modeling user exposure in recommendation. In *WWW*, 2016.

[9] Haoxuan Li, Chunyuan Zheng, Wenjie Wang, Hao Wang, Fuli Feng, and Xiao-Hua Zhou. Debiased recommendation with noisy feedback. In *SIGKDD*, 2024.

[10] Shuqiang Zhang, Yuchao Zhang, Jinkun Chen, and Haochen Sui. Addressing correlated latent exogenous variables in debiased recommender systems. In *SIGKDD*, 2025.

[11] Hao Wang, Zhichao Chen, Haotian Wang, Yanchao Tan, Licheng Pan, Tianqiao Liu, Xu Chen, Haoxuan Li, and Zhouchen Lin. Unbiased recommender learning from implicit feedback via weakly supervised learning. In *ICML*, 2025.

[12] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. Collaborative filtering with social exposure: A modular approach to social recommendation. In *AAAI*, 2018.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[14] Fengxiang He, Tongliang Liu, Geoffrey I Webb, and Dacheng Tao. Instance-dependent pu learning by bayesian optimal relabeling. *arXiv preprint arXiv:1808.02180*, 2018.

[15] Jessa Bekker, Pieter Robberechts, and Jesse Davis. Beyond the selected completely at random assumption for learning from positive and unlabeled data. In *ECML/PKDD*, 2019.

[16] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In *WSDM*, 2020.

[17] Chunyuan Zheng, Haocheng Yang, Haoxuan Li, and Mengyue Yang. Unveiling extraneous sampling bias with data missing-not-at-random. In *NeurIPS*, 2025.

[18] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*, 2016.

[19] Harald Steck. Training and testing of recommender systems on data missing not at random. In *SIGKDD*, 2010.

[20] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*, 2018.

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

[22] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In *WSDM*, 2020.

[23] Yuta Saito. Unbiased pairwise learning from biased implicit feedback. In *ICTIR*, 2020.

[24] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *AISTATS*, 2017.

[25] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[26] Yu Bai, Song Mei, Huan Wang, and Caiming Xiong. Don't just blame over-parametrization for over-confidence: Theoretical analysis of calibration in binary classification. In *ICML*, 2021.

[27] Haoxuan Li, Chunyuan Zheng, and Peng Wu. StableDR: Stabilized doubly robust learning for recommendation on data missing not at random. In *ICLR*, 2023.

[28] Hao Wang, Tai-Wei Chang, Tianqiao Liu, Jianmin Huang, Zhichao Chen, Chao Yu, Ruopeng Li, and Wei Chu. Escm2: entire space counterfactual multi-task model for post-click conversion rate estimation. In *SIGIR*, 2022.

[29] Haoxuan Li, Yan Lyu, Chunyuan Zheng, and Peng Wu. TDR-CL: Targeted doubly robust collaborative learning for debiased recommendations. In *ICLR*, 2023.

[30] Jun Wang, Haoxuan Li, Chi Zhang, Dongxu Liang, Enyun Yu, Wenwu Ou, and Wenjia Wang. Counterclr: Counterfactual contrastive learning with non-random missing data in recommendation. In *ICDM*, 2023.

[31] Yanghao Xiao, Haoxuan Li, Yongqiang Tang, and Wensheng Zhang. Addressing hidden confounding with heterogeneous observational datasets for recommendation. In *NeurIPS*, 2024.

[32] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. *NeurIPS*, 2018.

[33] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *ICML*, 2016.

[34] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[35] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *ICML*, 2017.

[36] Siyi Lin, Sheng Zhou, Jiawei Chen, Yan Feng, Qihao Shi, Chun Chen, Ying Li, and Can Wang. ReCRec: Reasoning the causes of implicit feedback for debiased recommendation. *ACM Transactions on Information Systems*, 2024.

[37] Yi Ren, Hongyan Tang, Jiangpeng Rong, and Siwen Zhu. Unbiased pairwise learning from implicit feedback for recommender systems without biased variance control. In *SIGIR*, 2023.

[38] Rong Pan and Martin Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *SIGKDD*, 2009.

[39] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*, 2016.

[40] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. In *AAAI*, 2020.

[41] Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. Unbiased implicit recommendation and propensity estimation via combinational joint learning. In *RecSys*, 2020.

[42] Jae-woong Lee, Seongmin Park, Joonseok Lee, and Jongwuk Lee. Bilateral self-unbiased learning from biased implicit feedback. In *SIGIR*, 2022.

[43] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *SIGIR*, 2020.

[44] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. Autodebias: Learning to debias for recommendation. In *SIGIR*, 2021.

[45] Honglei Zhang, Shuyi Wang, Haoxuan Li, Chunyuan Zheng, Xu Chen, Li Liu, Shanshan Luo, and Peng Wu. Uncovering the propensity identification problem in debiased recommendations. In *ICDE*, 2024.

[46] Hao Wang, Jiajun Fan, Zhichao Chen, Haoxuan Li, Weiming Liu, Tianqiao Liu, Quanyu Dai, Yichao Wang, Zhenhua Dong, and Ruiming Tang. Optimal transport for treatment effect estimation. In *NeurIPS*, 2023.

[47] Peng Wu, Haoxuan Li, Chunyuan Zheng, Yan Zeng, Jiawei Chen, Yang Liu, Ruocheng Guo, and Kun Zhang. Learning counterfactual outcomes under rank preservation. In *NeurIPS*, 2025.

[48] Hao Wang, Zhichao Chen, Zhaoran Liu, Xu Chen, Haoxuan Li, and Zhouchen Lin. Proximity matters: Local proximity enhanced balancing for treatment effect estimation. In *SIGKDD*, 2025.

[49] Quanyu Dai, Haoxuan Li, Peng Wu, Zhenhua Dong, Xiao-Hua Zhou, Rui Zhang, Xiuqiang He, Rui Zhang, and Jie Sun. A generalized doubly robust learning framework for debiasing post-click conversion rate prediction. In *SIGKDD*, 2022.

[50] Peng Wu, Haoxuan Li, Yuhao Deng, Wenjie Hu, Quanyu Dai, Zhenhua Dong, Jie Sun, Rui Zhang, and Xiao-Hua Zhou. On the opportunity of causal learning in recommendation systems: Foundation, estimation, prediction and challenges. In *IJCAI*, 2022.

[51] Haoxuan Li, Chunyuan Zheng, Sihao Ding, Fuli Feng, Xiangnan He, Zhi Geng, and Peng Wu. Be aware of the neighborhood effect: Modeling selection bias under interference for recommendation. In *ICLR*, 2024.

[52] Wenjie Wang, Yang Zhang, Haoxuan Li, Peng Wu, Fuli Feng, and Xiangnan He. Causal recommendation: Progresses and future directions. In *Tutorial on SIGIR*, 2023.

[53] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, and Peng Wu. Balancing unobserved confounding with a few unbiased ratings in debiased recommendations. In *WWW*, 2023.

[54] Chuan Zhou, Yaxuan Li, Chunyuan Zheng, Haiteng Zhang, Min Zhang, Haoxuan Li, and Mingming Gong. A two-stage pretraining-finetuning framework for treatment effect estimation with unmeasured confounding. In *SIGKDD*, 2025.

[55] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal intervention for leveraging popularity bias in recommendation. In *SIGIR*, 2021.

[56] Haoxuan Li, Kunhan Wu, Chunyuan Zheng, Yanghao Xiao, Hao Wang, Zhi Geng, Fuli Feng, Xiangnan He, and Peng Wu. Removing hidden confounding in recommendation: a unified multi-task learning approach. In *NeurIPS*, 2023.

[57] Haoxuan Li, Chunyuan Zheng, Shuyi Wang, Kunhan Wu, Eric Wang, Peng Wu, Zhi Geng, Xu Chen, and Xiao-Hua Zhou. Relaxing the accurate imputation assumption in doubly robust learning for debiased collaborative filtering. In *ICML*, 2024.

[58] Sami Khenissi and Olfa Nasraoui. Modeling and counteracting exposure bias in recommender systems. *arXiv preprint arXiv:2001.04832*, 2020.

[59] Haoxuan Li, Quanyu Dai, Yuru Li, Yan Lyu, Zhenhua Dong, Xiao-Hua Zhou, and Peng Wu. Multiple robust learning for recommendation. In *AAAI*, 2023.

[60] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, Peng Wu, and Peng Cui. Propensity matters: Measuring and enhancing balancing for recommendation. In *ICML*, 2023.

[61] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, Peng Wu, Zhi Geng, Xu Chen, and Peng Cui. Debiased collaborative filtering with kernel-based causal balancing. In *ICLR*, 2024.

[62] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *SIGKDD*, 2018.

[63] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning. In *WWW*, 2020.

[64] Hao Wang, Zhichao Chen, Zhaoran Liu, Haozhe Li, Degui Yang, Xinggao Liu, and Haoxuan Li. Entire space counterfactual learning for reliable content recommendations. *IEEE Transactions on Information Forensics and Security*, 2025.

[65] Tiankai Gu, Kun Kuang, Hong Zhu, Jingjie Li, Zhenhua Dong, Wenjie Hu, Zhenguo Li, Xiuqiang He, and Yue Liu. Estimating true post-click conversion via group-stratified counterfactual inference. In *SIGKDD*, 2021.

[66] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *SIGKDD*, 2021.

[67] Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems*, 42(1):1–27, 2023.

[68] Haoxuan Li, Chunyuan Zheng, Peng Wu, Kun Kuang, Yue Liu, and Peng Cui. Who should be given incentives? counterfactual optimal treatment regimes learning for recommendation. In *SIGKDD*, 2023.

[69] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NeurIPS*, 2007.

[70] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *NeurIPS*, 2000.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We list in the abstract and introduction the contributions of this paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We will provide a proof in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have stated the experimental details in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data is publicly available and we will provide all code if the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided all the details in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We include the error bars in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include the computer resources and approximate time of execution in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper has no ethics problem.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work is about counterfactual estimation under missing treatment, posing no direct societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There are no such risks in our paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original papers of the used assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The method in this paper does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proof of Theorem

Assumption A.1 (Inverse Representation and Function Class [35])**.** The representation $\Phi : \mathcal{X} \to \mathcal{A}$ is a one-to-one function, with inverse Ψ. Let G be a family of functions $g : \mathcal{A} \to \mathcal{Y}$. Assume there exists a constant $B_\Phi > 0$, such that $\frac{1}{B_\Phi} \cdot (h \circ \Phi \circ \Psi(a) - Y(1))^2 \in \mathrm{G}$.

Based on Assumption A.1, we then derive the following generalization bound:

Theorem A.2 (Generalization Bound)**.** *Under Assumption A.1, the deviation between the estimated relevance $h(\Phi(x))$ and expected relevance $m_1(x) = \mathbb{E}[Y(1) \mid X = x]$ averaging on all user-item pairs has the upper bound:*

$$\mathbb{E}_x[(h(\Phi(x)) - m_1(x))^2] \leq \underbrace{\mathbb{E}_{x|r,o}[(h(\Phi(x)) - Y(1))^2 \mid R = 1, O = 1]}_{\textit{factual loss of the DP and HE groups}}$$
$$+ \mathbb{P}(O = 0 \mid R = 1) \cdot B_\Phi \cdot \underbrace{\textit{IPM}_G(p_\Phi^{R=1,O=0}, p_\Phi^{R=1,O=1})}_{\textit{distribution shift on O given R=1}} + \underbrace{\mathbb{P}(R = 0)}_{\textit{UN group}} \cdot B_\Phi \cdot \underbrace{\textit{IPM}_G(p_\Phi^{R=0}, p_\Phi^{R=1})}_{\textit{distribution shift on R}}$$
$$- \underbrace{\mathbb{E}[(Y(1) - m_1(x))^2]}_{\textit{variance of potential outcome}}.$$

Proof. For simplicity, we define $\epsilon(x) = \mathbb{E}[(h(\Phi(x)) - Y(1))^2 \mid X = x]$ and $\epsilon_{Total} = \mathbb{E}_x[\epsilon(x)] = \mathbb{E}[(h(\Phi(x)) - Y(1))^2]$. The target error that we would like to bound is $\epsilon_{Target} := \mathbb{E}_x[(h(\Phi(x)) - m_1(x))^2]$. We will first derive the connection between the target error ϵ_{Target} and the total error ϵ_{Total}. Consider the following conditional expectation:

$$\begin{aligned}
&\mathbb{E}[(h(\Phi(x)) - Y(1))^2 \mid X = x] \\
=&\mathbb{E}[((h(\Phi(x)) - m_1(x)) + (m_1(x) - Y(1)))^2 \mid X = x] \\
=&\mathbb{E}[(h(\Phi(x)) - m_1(x))^2 \mid X = x] + \mathbb{E}[(m_1(x) - Y(1))^2 \mid X = x] \\
&+ 2\mathbb{E}[(h(\Phi(x)) - m_1(x)) \cdot (m_1(x) - Y(1)) \mid X = x] \\
=&\mathbb{E}[(h(\Phi(x)) - m_1(x))^2 \mid X = x] + \mathbb{E}[(m_1(x) - Y(1))^2 \mid X = x].
\end{aligned}$$

Taking expectation w.r.t the distribution of X, we have:

$$\begin{aligned}
&\epsilon_{Total} \\
=&\mathbb{E}_x\{\mathbb{E}[(h(\Phi(x)) - Y(1))^2 \mid X = x]\} \\
=&\mathbb{E}_x\{\mathbb{E}[(h(\Phi(x)) - m_1(x))^2 \mid X = x]\} + \mathbb{E}_x\{\mathbb{E}[(m_1(x) - Y(1))^2 \mid X = x]\} \\
=&\mathbb{E}[(h(\Phi(x)) - m_1(x))^2] + \mathbb{E}[(m_1(x) - Y(1))^2] \\
=&\epsilon_{Target} + \mathbb{E}[(m_1(x) - Y(1))^2].
\end{aligned} \tag{2}$$

Denoting $L(x) = (h(\Phi(x)) - Y(1))^2$, $v_0 = P(R = 0)$, $\epsilon^{R=1} = \mathbb{E}_x[\epsilon(x) \mid R = 1]$ and $\epsilon^{R=0} = \mathbb{E}_x[\epsilon(x) \mid R = 0]$, we can decompose ϵ_{Total} as follows:

$$\begin{aligned}
&\epsilon_{Total} \\
=&\mathbb{E}_{x|r}[\epsilon(x) \mid R = 1]P(R = 1) + \mathbb{E}_{x|r}[\epsilon(x) \mid R = 0]P(R = 0) \\
=&\epsilon^{R=1}(1 - v_0) + \epsilon^{R=0}v_0 \\
=&\epsilon^{R=1} - \epsilon^{R=1}v_0 + \epsilon^{R=0}v_0 \\
=&\epsilon^{R=1} + v_0(\epsilon^{R=0} - \epsilon^{R=1}) \\
=&\epsilon^{R=1} + v_0\left(\int \epsilon(x)p(x \mid R = 0)dx - \int \epsilon(x)p(x \mid R = 1)dx\right) \\
=&\epsilon^{R=1} + v_0 \int \epsilon(x)(p(x \mid R = 0) - p(x \mid R = 1))dx \\
=&\epsilon^{R=1} + v_0 \int \mathbb{E}[(h(\Phi(x)) - Y(1))^2 \mid X = x](p(x \mid R = 0) - p(x \mid R = 1))dx \\
=&\epsilon^{R=1} + v_0 \int L(x)(p(x|R = 0) - p(x|R = 1))dx.
\end{aligned} \tag{3}$$

Based on Assumption A.1, we have $L(x)/B_\Phi = \frac{1}{B_\Phi}(h \circ \Psi \circ \Psi(a) - Y(1))^2 \in G$, and thus

$$\begin{aligned}
&\int L(x)(p(x|R=0) - p(x|R=1))dx \\
=&B_\Phi \int \frac{L(x)}{B_\Phi}(p(x|R=0) - p(x|R=1))dx \\
=&B_\Phi \int \frac{L(\Psi(a))}{B_\Phi}(p(a|R=0) - p(a|R=1))da \\
\leq&B_\Phi \cdot \sup_{g \in G} |\int g(a)(p_\Phi^{R=0}(a) - p_\Phi^{R=1}(a))da| \\
=&B_\Phi \cdot IPM_G(p_\Phi^{R=0}, p_\Phi^{R=1}).
\end{aligned} \tag{4}$$

Combining Eq. (3) and Eq. (4), we have

$$\epsilon_{Total} \leq \epsilon^{R=1} + v_0 \cdot B_\Phi \cdot IPM_G(p_\Phi^{R=0}, p_\Phi^{R=1}). \tag{5}$$

Next, we denote $u_0 = P(O = 0 \mid R = 1), \epsilon^{R=1,O=1} = \mathbb{E}_{x|r,o}[\epsilon(x) \mid R = 1, O = 1]$ and $\epsilon^{R=1,O=0} = \mathbb{E}_{x|r,o}[\epsilon(x) \mid R = 1, O = 0]$ and then decompose $\epsilon^{R=1}$ as follows:

$$\begin{aligned}
&\epsilon^{R=1} \\
=&\epsilon^{R=1,O=1} \cdot (1 - u_0) + \epsilon^{R=1,O=0} \cdot u_0 \\
=&\epsilon^{R=1,O=1} + u_0 \cdot (\epsilon^{R=1,O=0} - \epsilon^{R=1,O=1}) \\
=&\epsilon^{R=1,O=1} + u_0 \left(\int \epsilon(x) p(x \mid R=1, O=0)dx - \int \epsilon(x) p(x \mid R=1, O=1)dx \right) \\
=&\epsilon^{R=1,O=1} + u_0 \int \epsilon(x)(p(x \mid R=1, O=0) - p(x \mid R=1, O=1))dx \\
=&\epsilon^{R=1,O=1} \\
&+ u_0 \int \mathbb{E}[(h(\Phi(x)) - Y(1))^2 \mid X = x](p(x \mid R=1, O=0) - p(x \mid R=1, O=1))dx \\
=&\epsilon^{R=1,O=1} + u_0 \int L(x)(p(x \mid R=1, O=0) - p(x \mid R=1, O=1))dx.
\end{aligned} \tag{6}$$

Analogous to the derivation of Eq. (4), we have

$$\begin{aligned}
&\int L(x)(p(x \mid R=1, O=0) - p(x \mid R=1, O=1))dx \\
=&B_\Phi \int \frac{L(x)}{B_\Phi}(p(x \mid R=1, O=0) - p(x \mid R=1, O=1))dx \\
=&B_\Phi \int \frac{L(\Psi(a))}{B_\Phi}(p(a \mid R=1, O=0) - p(a \mid R=1, O=1))da \\
\leq&B_\Phi \cdot \sup_{g \in G} |\int g(a)(p_\Phi^{R=1,O=0}(a) - p_\Phi^{R=1,O=1}(a))da| \\
=&B_\Phi \cdot IPM_G(p_\Phi^{R=1,O=0}, p_\Phi^{R=1,O=1}).
\end{aligned} \tag{7}$$

Combining Eq. (6) and Eq. (7), we have

$$\epsilon^{R=1} \leq \epsilon^{R=1,O=1} + u_0 \cdot B_\Phi \cdot IPM_G(p_\Phi^{R=1,O=0}, p_\Phi^{R=1,O=1}). \tag{8}$$

Note that $\epsilon^{R=1,O=1} = \mathbb{E}_{x|r,o}[\epsilon(x) \mid R = 1, O = 1] = \mathbb{E}_{x|r,o}[(h(\Phi(x)) - Y(1))^2 \mid R = 1, O = 1]$, and combining the results of Eq. (2), Eq. (5) and Eq. (8), we complete the proof of the theorem.

□

B Data Preprocess

All of the datasets used in this paper contain user ratings on an item as explicit feedback. So we simulate the implicit feedback mechanism using the following data preprocessing pipeline.

1. Transform all ratings into relevance scores using the following formula:

$$\gamma_{u,i} = \epsilon + (1-\epsilon)\frac{2r_{u,i}-1}{2r_{\text{max}}-1},$$

   where $r_{u,i}$ denotes the rating for each user-item pair in the observed set $\mathcal{O}$, and r_{max} is the maximum rating. The parameter $\epsilon \in [0,1]$ controls the noise level. Following the previous studies [23, 36], we set $\epsilon = 0.1$ for the training datasets and $\epsilon = 0$ for the test datasets to ensure unbiased evaluation.

2. Sample the binary relevance $S_{u,i}$ using Bernoulli sampling:

$$S_{u,i} \sim \text{Bern}(\gamma_{u,i}), \quad \forall (u,i) \in \mathcal{O},$$

   where $\text{Bern}(\cdot)$ denotes the Bernoulli distribution.

3. Define the exposure variable for all user-item pair $(u,i) \in \mathcal{U} \times \mathcal{I}$:

$$O_{u,i} = \begin{cases} 1 & \text{if item } i \text{ is rated by user } u, \\ 0 & \text{if item } i \text{ is not rated by user } u. \end{cases}$$

4. Finally, we sample the binary outcome as the implicit feedback:

$$Y_{u,i} = \begin{cases} S_{u,i} & \text{if } O_{u,i} = 1, \\ 0 & \text{if } O_{u,i} = 1. \end{cases}$$

   Note that $S_{u,i}$ and $O_{u,i}$ are unobservable in our setting and the training data is $\{(u,i,Y_{u,i}) : (u,i) \in \mathcal{U} \times \mathcal{I}\}$.

C Efficient Alternatives of Kernel-Based Hypersphere Model

Although the kernel-based hypersphere model in Section 3.3 introduces $O(m^2)$ complexity with m positive samples, we would like to emphasize that the calculation of the kernel function is only one possible option; the core lies in the framework of treatment imputation with confidence. Here, we provide two more efficient and scalable kernel function approximation methods, i.e., Random Fourier Features (RFF) [69] and Nyström approximation [70] to replace the method in Section 3.3. Experimental results on the Coat and Yahoo datasets are presented in 4, showing that regardless of the kernel function calculation method adopted, the ranking performance is guaranteed.

Table 4: Ranking performance with efficient alternatives of kernel-based hypersphere model.

	Methods	**K=3**			**K=5**			**K=8**		
		NDCG@K	**Recall@K**	**MAP@K**	**NDCG@K**	**Recall@K**	**MAP@K**	**NDCG@K**	**Recall@K**	**MAP@K**
Yahoo	Ours (SVDD)	0.562 ± 0.007	0.624 ± 0.009	0.499 ± 0.007	0.625 ± 0.005	0.776 ± 0.008	0.547 ± 0.006	0.681 ± 0.004	0.930 ± 0.004	0.582 ± 0.005
	Ours (RFF)	$\mathbf{0.578 \pm 0.003}$	$\mathbf{0.645 \pm 0.005}$	$\mathbf{0.513 \pm 0.003}$	$\mathbf{0.639 \pm 0.003}$	$\mathbf{0.790 \pm 0.003}$	$\mathbf{0.560 \pm 0.003}$	$\mathbf{0.693 \pm 0.002}$	$\mathbf{0.939 \pm 0.002}$	0.585 ± 0.003
	Ours (Nyström)	0.572 ± 0.007	0.637 ± 0.005	0.508 ± 0.008	0.634 ± 0.005	0.786 ± 0.004	0.555 ± 0.007	0.685 ± 0.006	0.925 ± 0.004	$\mathbf{0.587 \pm 0.007}$
Coat	Ours (SVDD)	0.368 ± 0.011	0.382 ± 0.012	0.296 ± 0.009	0.414 ± 0.012	0.478 ± 0.014	0.332 ± 0.009	0.473 ± 0.012	0.660 ± 0.021	0.369 ± 0.009
	Ours (RFF)	0.361 ± 0.012	0.384 ± 0.022	0.282 ± 0.015	0.387 ± 0.005	0.454 ± 0.030	0.309 ± 0.010	0.468 ± 0.008	$\mathbf{0.671 \pm 0.036}$	0.360 ± 0.011
	Ours (Nyström)	$\mathbf{0.389 \pm 0.011}$	$\mathbf{0.390 \pm 0.017}$	$\mathbf{0.314 \pm 0.012}$	$\mathbf{0.427 \pm 0.023}$	$\mathbf{0.495 \pm 0.038}$	$\mathbf{0.347 \pm 0.019}$	$\mathbf{0.486 \pm 0.015}$	0.653 ± 0.030	$\mathbf{0.384 \pm 0.013}$

D Experiments Compute Resources

We conduct all experiments on a server with 112-core Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz. The server is equipped with a 512GB random access memory (RAM). To reproduce all the experimental results including the baselines takes a few hours.