Debiasing Learning based Cross-domain Recommendation

Siqing Li[1,4], Liuyi Yao[3], Shanlei Mu[1,4], Wayne Xin Zhao[2,4,5*], Yaliang Li[3],
Tonglei Guo[3], Bolin Ding[3] and Ji-Rong Wen[1,4]
[1]School of Information, Renmin University of China
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Alibaba Group
[4]Beijing Key Laboratory of Big Data Management and Analysis Methods
[5]Beijing Academy of Artificial Intelligence
{lisiqing,slmu,jrwen}@ruc.edu.cn,batmanfly@gmail.com,
{yly287738,yaliang.li,tonglei.gtl,bolin.ding}@alibaba-inc.com

ABSTRACT

As it becomes prevalent that user information exists in multiple platforms or services, cross-domain recommendation has been an important task in industry. Although it is well known that users tend to show different preferences in different domains, existing studies seldom model how domain biases affect user preferences. Focused on this issue, we develop a casual-based approach to mitigating the domain biases when transferring the user information cross domains. To be specific, this paper presents a novel debiasing learning based cross-domain recommendation framework with causal embedding. In this framework, we design a novel Inverse-Propensity-Score (IPS) estimator designed for cross-domain scenario, and further propose three kinds of restrictions for propensity score learning. Our framework can be generally applied to various recommendation algorithms for cross-domain recommendation. Extensive experiments on both public and industry datasets have demonstrated the effectiveness of the proposed framework.

CCS CONCEPTS

• **Information systems → Collaborative filtering**.

KEYWORDS

Cross-domain Recommendation; Debiasing Learning; Causal Embedding

ACM Reference Format:
Siqing Li, Liuyi Yao, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Tonglei Guo, Bolin Ding and Ji-Rong Wen. 2021. Debiasing Learning based Cross-domain Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3447548.3467067

*Corresponding author.

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

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467067

1 INTRODUCTION

Nowadays, recommender system plays an important role in e-commerce platforms, such as Amazon and Taobao. Traditional collaborative filtering methods recommend items to users based on their historical rating or click behaviors in a single domain [9, 24, 37, 47]. With the development of e-commerce platform business, users may have behaviors in different domains. For example, in Taobao, there are multiple business domains with different domain settings, containing *Gouwuche* (purchase-guided domain setting), *Shoucai* (exploration-guided domain setting), and so on. As the variety of web services has increased, user information can be obtained from their activities in other services. Therefore, cross-domain recommender system has gained research attention in recent years.

Compared with single-domain scenarios, the major challenge of cross-domain recommendation is that the users in different domains tend to have varied behavioral patterns [10] or different selection preferences [48]. Therefore, it is difficult to transfer useful information across domains and learn a comprehensive user preference. In the literature, various methods have been proposed for tackling this challenge [3, 29, 40]. Existing methods mainly focus on how to learn effective information representations that are transferable across domains. The basic idea is to bridge the semantic gap between different domains with shared parameters [3, 40], feature mapping [10] or semantic space alignment.

Figure 1: Examples of two items with different tastes purchased by a Taobao user in two domains.

Although these methods have largely improved the performance of cross-domain recommendation, they have not explicitly modeled why and how user preferences change across domains. Consequently, it is unable to accurately capture real user preference *in general* or *with respect to a specific domain.* To see this, Figure 1 presents a real user in Taobao with her click behaviors on the applications of Gouwuche and Shoucai (*i.e.,* two domains). The user has exposed (in public profile) her own general preferences "Liquid foundation sold by Sephora" about cosmetics. However, she has different behaviors in the two domains. In Gouwuche, she clicked a Maybelline's cheap foundation (the left figure), while in Shoucai, she clicked a Chanel's expensive foundation (the right figure). We argue that such preference differences are caused by underlying domain bias, *i.e.,* user's sense of item attributes may vary due to domain factors. As introduced before, Gouwuche (purchase-guided domain setting) and Shoucai (exploration-guided domain setting) indeed have special marketing strategies and target on different scenarios. To solve the above issue, the core problem becomes *how to mitigate the biases yielded in current domain when transferring the user information cross domains to improve recommendation?*

From the causal perspective, the domain biases are caused by the existence of the domain-specific confounders, which are the variables/factors affecting both the user preference and the user behavior. In the example of Figure 1, domain setting can be considered as a latent domain-specific confounder. The domain-specific confounders bring two kinds of biases: user preference bias and data selection bias [39]. The user preference is biased as it is directly affected by the domain-specific confounders. For example, in Gouwuche, the user's general preference "liquid foundation" is changed to "Maybelline's cheap liquid foundation", due to the effect of its purchase-guided domain setting. Then the preference bias further causes the data selection bias, which means most user-item interactions in the observed data are related to user-preferred items in the specific domain. For example, in Gouwuche, most of the interactions in observed dataset are affordable items. While, in Shoucai domain, most of them are gorgeous and fashionable items. It would incorporate such domain bias if directly transferring the behaviors in Gouwuche to Shoucai.

In light of the above causal view, we propose a novel propensity score based cross-domain framework. Propensity score, which denotes the user preference strength to an item [38], is widely adopted to solve the bias in single domain by re-weighting each transaction in the observed data. We first design a novel Inverse-Propensity-Score (IPS) estimator, which generalizes the traditional IPS estimator to the cross-domain scenarios. Due to the difficulty in estimating the propensity score via statistical methods in cross-domain scenario, we design three restrictions to learn the propensity score. Our approach can effectively eliminate the domain biases, and characterize user preference in a more accurate way. Overall, our contributions are that: (1) Correct the data selection bias in cross-domain scenarios by generalized propensity score; (2) Propose a novel way to estimate the propensity score when domain-specific confounders are unobserved; (3) Model the user preference shift in different domains by propensity score to handle preference bias.

With this general framework, various recommendation methods can be extended for cross-domain recommendation.

To the best of our knowledge, it is the first time that debiasing learning has been applied to cross-domain recommendation. To validate the effectiveness of the proposed framework, we conduct a series of experiments on both public dataset from Amazon and industry dataset from Taobao. Experimental results under different metrics confirm that the proposed framework outperforms the state-of-the-art methods for cross-domain recommendation.

2 RELATED WORK

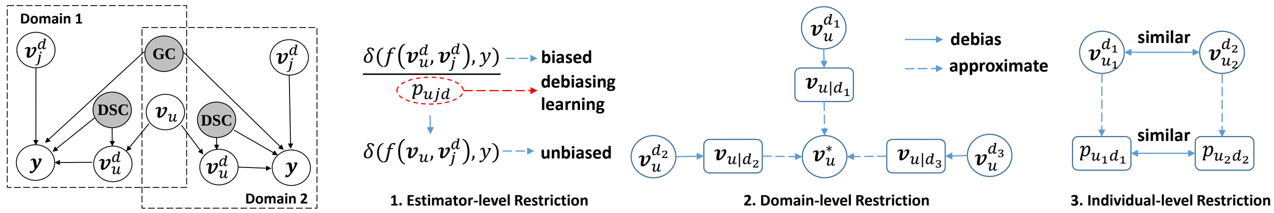
In this section, we discuss related work that are close to our work.

Cross-domain Recommendation. Cross-domain recommendation task aims to leverage information from other domains to help the recommendation in target domain. Traditional methods using collaborative filtering (CF) can be divided into two categories, namely aggregating the knowledge among domains and transferring knowledge from source domain to target domain. In the first category, most of them conduct their research based on matrix factorization (MF) [3, 40]. Furthermore, some researchers consider that different users in different domains have different behavior patterns [4, 26, 29, 44]. So they jointly learn the shared cross-domain features and domain-specific features by using LDA [4], probabilistic factorization generative model [44] and social network [26]. Some researchers propose to transfer knowledge from source domain to target domain by utilizing the overlapped part as a bridge to link the domains [20]. Since traditional methods have difficulties in learning complex user-item interactions, deep learning based methods have been proposed [10, 11, 16, 25, 31]. Among them, user privacy issue has been considered [11] by only using the item information for cross-domain recommendation. They learn the item embeddings in source domain by CF and transfer them to target domain with attention mechanism. These works only reduced the influence of the source domain during the fine-tuning of the target domain, but did not accurately determine the confounding factors contained in the migrated information.

Debias in Recommender System. Although recent years have witnessed the rapid growth of works on recommender systems (RS), most studies focus on designing machine learning models to better adapt to user behavior data. However, user behavior data is observational. This leads to widespread bias in the data, including selection bias [15] in explicit feedback data, position bias [21] and exposure bias [35] in implicit feedback data and popularity bias [1]. A large number of methods have been proposed to eliminate the biases. Some researchers propose to mitigate selection bias by propensity score [39], ATOP [41] and data imputation [8, 15, 32, 33]. Exposure bias can be mitigated in evaluation [28], model training [30, 42, 46] and sampling [6, 43]. Recently, some researchers provide a survey [7] summarizing seven types of biases in recommendation, along with their definitions and characteristics. However, all of these works do not consider the biases that are produced in cross-domain scenario and how to mitigate the impact of domains.

3 PRELIMINARY

In this section, we first introduce the task definition of cross-domain recommendation, and then introduce the background about causal.



(a) Causal graph. (b) Three restrictions in our method.

Figure 2: The causal graph and three restrictions in our proposed method. (DSC means domain-specific confounder, *e.g.,* purchase-guided marketing strategy. GC means general confounder, *e.g.,* the display position.)

3.1 Problem Definition

Assume that we have a set of domains, users and items, denoted by $\mathcal{D}$, $\mathcal{U}$ and $\mathcal{I}$ respectively, where $d \in \mathcal{D}$ denotes a domain, $u \in \mathcal{U}$ denotes a user and $i \in \mathcal{I}$ denotes an item. The numbers of domains, users and items are denoted as D, U and I. In our setting, there exist a number of overlapped users. While, we do not make specific assumption about overlapped items. Instead, the items from different domains share the same attribute set $\mathcal{J}$. Formally, each item i is associated with attributes $\mathcal{J}_i = \{j_1, ..., j_m\}$, where $\mathcal{J}_i \subset \mathcal{J}$. For example, an item has attributes containing category, seller, brand and price. Given a user u, suppose there are two kinds of preference representations (k-dimensional vectors), namely *general preference* $\boldsymbol{v}_u \in \mathbb{R}^k$, which is shared among different domains, and *domain-specific preference* $\boldsymbol{v}_u^d \in \mathbb{R}^k$, which is affected by domain-related factors.

Based on the above notations, we now define the task of cross-domain recommendation task. Formally, given the user u and the attributes $\mathcal{J}_i$ of each item i and its corresponding domain d, we aim to learn the user's general preference $\boldsymbol{v}_u$ debiased from the preference that is affected by domain $\boldsymbol{v}_u^d$ and better predict the user's behaviour y_{ui}. And the debiased $\boldsymbol{v}_u$ can be transferred to other domains to help domain-specific recommendation.

3.2 Causal Background

Causal Graph. A causal graph, denoted as $\mathcal{G}$, is a directed acyclic graph (DAG) which describes the causal relationships between variables [36]. $\mathcal{G} = \langle \mathcal{R}, \mathcal{E} \rangle$ where $\mathcal{R}$ is a set of random variables and $\mathcal{E}$ is the set of edges with each edge $r_i \rightarrow r_j$ representing i-th variable r_i is a direct cause of j-th variable r_j.

Propensity Score. In causal inference, the propensity score is defined as the probability of an individual being assigned to a specific treatment [38]. In the recommender system, the treatment can be viewed as being given one specific item. Then, the propensity score is the marginal probability of observing a user's rating or clicking behavior on a certain item [39], which to a certain degree, reflects the user's preference strength to this item. In single-domain recommendation, propensity score is commonly adopted to handle the selection bias. Specifically, each observation is weighted by its inverse propensity to obtain the unbiased estimation of performance measure [7, 39, 45]. To model cross-domain preference bias, we introduce the *propensity score* p_{ujd}, which can be viewed as a preference degree of user u for attribute j in domain d. It captures domain-related confounders to a certain extent.

Cross-domain Bias. The cross-domain biases, including user preference bias and data selection bias, are caused by the domain-specific confounders in the cross-domain recommender system. In causal inference, the *confounder*, narrowly speaking, refers to the variables that are the common causes of two variables [18]. In our scenario, they are the variables that affect both the user preference and the user behavior. As the example in Figure 1 shows, the domain setting is a latent domain-specific confounder, where purchase-guided domain setting and exploration-guided domain setting are adopted by Gouwuche and Shoucai, respectively. Influenced by different domain settings, the sample user adapts general preference "liquid foundation" to "Maybelline's cheap foundation" or "Chanel's expensive foundation" in the two domains accordingly. Such preference shifts directly brought by the domain-specific confounders are named as *preference bias.* The preference shift further causes the data selection bias, as users select products according to different preferences across domains.

4 METHODOLOGY

In this section, we present our debiasing learning framework with causal embedding for the cross-domain recommendation task.

4.1 Overview

We first construct a causal graph to analyze user behavior in Figure 2. User behaviors in recommender system are biased, which is caused by various confounders, including domain-specific confounders (DSC) and general confounders (GC). Domain-specific confounders, such as the purchase-guided marketing strategy, only affect the user preference in one specific domain. In contrast, general confounders, such as the display position, are domain-shared confounders that have influence to all domains. In our case, we focus on solving the *cross-domain selection bias* brought by domain-specific confounders. Because we solve the task from the perspective of causal, the embedding of users and item attributes are also called causal embedding. In order to learn the cross-domain confounders, consider such a scenario: regard the item attribute j in domain d given to user u as a treatment, and whether the user likes it as an outcome $\hat{y}$. We aim to model the confounders in the treatment and eliminate the biases in cross-domain learning.

We first propose a novel Inverse-Propensity-Score (IPS) estimator in Section 4.2, which generalizes the traditional IPS estimator [27] to the cross-domain scenarios. Furthermore, to address the difficulty of propensity score estimation in our cross-domain scenarios, three restrictions are proposed to estimate the propensity score in Section 4.3. Then we will introduce our training process and some discussion in Section 4.4.

4.2 IPS Estimator for Cross-domain Debiasing

In this part, we propose a novel Inverse-Propensity-Score (IPS) estimator, which generalizes the traditional IPS estimator to the cross-domain scenarios. We first introduce how to estimate the recommendation accuracy on all domains with user's information and item's attribute information. Then we present the proposed novel Inverse-Propensity-Score (IPS) estimator in cross-domain scenarios.

4.2.1 Recommendation Accuracy Estimator. In recommender system, the ideal measure of evaluating how well a predicted $\hat{Y}$ reflects the true ratings or click through rate in Y is:

$$\begin{aligned} R(\hat{Y}) &= \frac{1}{U \cdot J \cdot D} \sum_{u=1}^{U} \sum_{j=1}^{J} \sum_{d=1}^{D} \delta(\hat{y}, y) \\ &= \frac{1}{U \cdot J \cdot D} \sum_{u=1}^{U} \sum_{j=1}^{J} \sum_{d=1}^{D} \delta(f(\boldsymbol{v}_u, \boldsymbol{v}_j^d), y), \end{aligned} \tag{1}$$

where $\boldsymbol{v}_u$ is the user embedding and $\boldsymbol{v}_j^d$ is the embedding of the j-th attribute of item in domain d, $f(\cdot)$ is the recommendation method like Factorization Machine (FM) or DeepFM, and $\delta_{u,i}(y, \hat{y})$ is the error measure, which can be Mean Absolute Error (MAE), Mean Squared Error (MSE), or Binary Cross Entropy (BCE).

It's worth noting that, the above the measure is unable to calculate directly due to the missing data. The conventional practice is to estimate $R(\hat{Y})$ using the average over only the observed data:

$$\hat{R}_{\text{naive}}(\hat{Y}) = \frac{1}{|O_{data}|} \sum_{\langle u,j,d \rangle \in O_{data}} \delta(f(\boldsymbol{v}_u, \boldsymbol{v}_j^d), y), \tag{2}$$

where $O_{data} = \{\langle u, j, d \rangle : T_{ujd} = 1\}$ and $T_{ujd} = 1$ indicates $\langle u, j, d \rangle$ is observed. Following [39], we call this the *naive estimator*.

However, $\hat{R}_{\text{naive}}(\hat{Y})$ is not an unbiased estimate of the true performance $R(\hat{Y})$ because of the selection bias. As shown in Figure 2, the confounders, *i.e.,* domain-specific confounder DSC and general confounder GC, affect both the users' preferences and the ratings/clicking behaviors in different domains. Therefore, the observed data suffers from selection bias.

4.2.2 Proposed Novel IPS Estimator. To address the selection bias, we propose our designed Inverse-Propensity-Scoring estimator for cross-domain recommendation task. According to [19], the key to handling selection bias is to understand the process of generating observation patterns, which is typically called the *Assignment Mechanism* in causal inference. Following [39], we assume that the assignment mechanism is probabilistic. We set $p_{ujd} = P(T_{ujd} = 1)$ as the propensity score, which is the probability of occurrence introduced in Section 3. Then the Inverse-Propensity-Scoring (IPS) estimator is defined as:

$$\hat{R}_{\text{IPS}}(\hat{Y}|P) = \frac{1}{U \cdot J \cdot D} \sum_{\langle u,j,d \rangle \in O_{data}} \frac{\delta(f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d), y)}{p_{ujd}}, \tag{3}$$

Different from the naive estimator $\hat{R}_{naive}(\hat{Y})$, the IPS estimator is unbiased for any probabilistic assignment mechanism:

$$\begin{aligned} &E_T[\hat{R}_{\text{IPS}}(\hat{Y}|P)] \\ &= \frac{1}{U \cdot J \cdot D} \sum_{u} \sum_{j} \sum_{d} E_{T_{ujd}}\left[\frac{\delta(f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d), y)}{p_{ujd}} T_{ujd}\right] \\ &= \frac{1}{U \cdot J \cdot D} \sum_{u} \sum_{j} \sum_{d} \delta(f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d), y) = R(\hat{Y}). \end{aligned} \tag{4}$$

Consequently, the proposed IPS estimator can eliminate the selection bias during the training process.

4.3 Cross-domain Propensity Score learning

In previous studies of learning propensity score [39], the propensity score is estimated via statistical methods based on the observed factors related to user preference to a specific item. However, in our task, the user preference is biased in different domains and few of the domain factors that related to appearance of a user-item interaction in a certain domain are observed. Furthermore, as the user-item interaction expands into three dimensions of *user-item-domain*, the data sparsity problem becomes more serious. As a result, the number of missing data increases, leading to more inaccurate estimates of the propensity score. Therefore, it is difficult to estimate the propensity score in the same statistical way [39].

To overcome this challenge, we notice that propensity score has multiple meanings: it is the weighting parameter for debiasing in the estimator; meanwhile, by the definition of the propensity score, it also indicates the user preference. Accordingly, we design three restrictions to reduce the uncertainty in estimator level, domain level, and individual level, respectively. The three restrictions estimate the propensity score by exploring the role of propensity as the weighting parameter in estimator level and as the indicator of user preference in the domain and individual level.

4.3.1 Restriction in Estimator Level. In the estimation, the propensity score serves as the weighting parameter for debiasing. After debiasing by inverse propensity score weighting, the biased estimation is equal to the unbiased estimation. Motivated by this fact, we have:

$$\frac{\delta(f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d), y)}{p_{ujd}} = \delta(f(\boldsymbol{v}_u, \boldsymbol{v}_j^d), y), \tag{5}$$

where $\boldsymbol{v}_u$ represents user's general preferences (shared among different domains), p_{ujd} is the propensity score (a preference degree of different users for different attributes of item in different domains), and $\boldsymbol{v}_u^d$ represents user preferences that have changed after being affected by domain-related factors.

According to Eq. (5), the restriction of the propensity score p_{ujd} is:

$$p_{ujd} = \text{argmin}_p \sum_{d=1}^{|\mathcal{D}|} \lambda_d \left|\left| \frac{\delta(f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d), y)}{p_{ujd}} - \delta(f(\boldsymbol{v}_u, \boldsymbol{v}_j^d), y) \right|\right|_2^2, \tag{6}$$

where λ_d is the weight of domain d. Since $\delta_{u,i}(y,\hat{y})$ and $f(\boldsymbol{v}_u,\boldsymbol{v}_j)$ can be complicated, it is difficult to calculate the relationship between $\boldsymbol{v}_u$ and $\boldsymbol{v}_u^d$. So we use neural networks to approximate the relationship between $\boldsymbol{v}_u$ and $\boldsymbol{v}_u^d$:

$$\boldsymbol{v}_u = \mathrm{ReLU}(\boldsymbol{W}_3\mathrm{ReLU}(\boldsymbol{W}_1\boldsymbol{v}_u^d + \boldsymbol{w}_2 p_{ujd} + b1) + b2), \tag{7}$$

where $\boldsymbol{W}_1, \boldsymbol{w}_2, \boldsymbol{W}_3, b_1, b_2$ are trainable parameters. In other words, we recover the user's general preferences $\boldsymbol{v}_u$ by the nonlinear combination of the domain related preference $\boldsymbol{v}_u^d$ and the propensity score p_{ujd}.

Overall, when p_{ujd} is viewed as the weighting parameter for debiasing, the associated loss is:

$$Loss_1 = \sum_{d=1}^{|\mathcal{D}|} \lambda_d || \frac{\delta(f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d), y)}{p_{ujd}} - \delta(f(\boldsymbol{v}_u, \boldsymbol{v}_j^d), y)||_2^2. \tag{8}$$

4.3.2 Restriction in Domain Level. At domain level, the propensity score, combined with domain-specific user embedding $\boldsymbol{v}_u^d$, can recover the the general preference, as shown in *Eq.* (7). In this manner, we can recover the general preference in each of the $|\mathcal{D}|$ domains. Let $\boldsymbol{v}_{u|d}$ denote the estimation of general preference for user u for domain d (recovered by the propensity score p_{ujd} and domain-specific user embedding $\boldsymbol{v}_u^d$). Therefore, we can derive $|\mathcal{D}|$ estimations for general preference: $\{\boldsymbol{v}_{u|d}\}_{d=1}^{|\mathcal{D}|}$. Naturally, the general preferences among different domains should be similar. Therefore, motivated by the confounder balancing strategy in causal inference [2, 5, 13, 17], we propose to learn the propensity score by balancing the user's general preferences among different domains:

$$p_{ujd} = \mathrm{argmin}_p \sum_{d=1}^{|\mathcal{D}|} [\alpha_d \parallel \boldsymbol{v}_u^* - \boldsymbol{v}_{u|d} \parallel_2^2], \tag{9}$$

where the confounder weights $\{\alpha_d\}_{d=1}^{|\mathcal{D}|}$ is the inverse of sample size ratio, controlling the degree of the effect of the confounders in cross-domain, and $\boldsymbol{v}_u^*$ is the mean of $\boldsymbol{v}_{u|d}$, which is defined as $\boldsymbol{v}_u^* = 1/|\mathcal{D}| \sum_{d=1}^{|\mathcal{D}|} \boldsymbol{v}_{u|d}$. Restricting $\{\boldsymbol{v}_{u|d}\}_{d=1}^{|\mathcal{D}|}$ similar to each other is equivalent to making each of them close to their mean.

In a nutshell, the loss associated with the restriction at domain level is:

$$Loss_2 = \sum_{d=1}^{|\mathcal{D}|} [\alpha_d \parallel \boldsymbol{v}_u^* - \boldsymbol{v}_{u|d} \parallel_2^2]. \tag{10}$$

After multiple iterations, the user preferences affected by the domain can be separated from the user's static preferences, and only the user preferences shared by the domains can be transferred across domains.

4.3.3 Restriction in Individual Level. Since propensity score p_{ujd} represents user u's preference for attribute j in domain d, the closer the embeddings of two users in different domains are, the closer the preferences of two users will be. Therefore, we use Laplacian regularization to restrict on p_{ujd} in individual level:

$$Loss_L = \boldsymbol{p}^\top \boldsymbol{L}\boldsymbol{p} = \sum_{d_1 \neq d_2} \sum_{u_1 \neq u_2} \cos(\boldsymbol{v}_{u_1}^{d_1}, \boldsymbol{v}_{u_2}^{d_2}) ||\boldsymbol{p}_{u_1 d_1} - \boldsymbol{p}_{u_2 d_2}||_2^2, \tag{11}$$

where $\boldsymbol{p}_{ud}$ represents the user u's preference in domain d, $\boldsymbol{p}_{ud} = [p_{u1d}, p_{u2d}, ..., p_{u|\mathcal{J}|d}]$, and we use the cosine similarity to measure the closeness between user embeddings. The larger the $cos(\boldsymbol{v}_{u_1}^{d_1}, \boldsymbol{v}_{u_2}^{d_2})$, which means the smaller the distance between $\boldsymbol{v}_{u_1}^{d_1}$ and $\boldsymbol{v}_{u_2}^{d_2}$, and the closer the $\boldsymbol{p}_{u_1 d_1}$ and $\boldsymbol{p}_{u_2 d_2}$ should be. In practice, we do not need to examine all the pairs of users. For a target user, we can only consider top similar users with him/her.

4.4 Training and Discussion

In this part, we introduce the training process and present some discussion and analysis.

4.4.1 Training. The parameters to learn in our model include the parameters for the recommendation function in $f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d)$ denoted by Θ_f and debiasing parameters $\{p_{ujd}, \boldsymbol{W}_1, \boldsymbol{w}_2, \boldsymbol{W}_3, b_1, b_2\}$ denoted as Θ_d. We learn Θ_f by the proposed Inverse-Propensity-Scoring Estimator in *Eq.* (3). The propensity score and parameters in *Eq.* (7) are learned by integrating the three loss terms in *Eq.* (8), *Eq.* (10) and *Eq.* (11):

$$L = Loss_1 + Loss_2 + Loss_L. \tag{12}$$

In each iteration, we first train the Θ_f in each domain and then train the Θ_d across domains. The whole method is trained with back-propagation. The training algorithm is summarized in Appendix A.1. We will describe more implementation details in Appendix B.

4.4.2 Discussion. In this part, we present some discussions related to the proposed framework.

Generalizability. Notice that we propose a framework for cross-domain recommendation task to debias the domain-specific confounders. In our approach, the single-domain recommendation method $f(\boldsymbol{v}_u, \boldsymbol{v}_j)$ (Eq. (3)) can be instantiated with different methods for modeling the interaction between users and item attributes. We can also utilize this framework to solve different types of recommendation tasks by changing the evaluation measure $\delta_{u,i}(y,\hat{y})$ (Eq. (3)) accordingly, *e.g.,* rating prediction or click-through-rate prediction.

Method Analysis. In the cause and effect diagram, we hope to find the relationship among user's general preferences $\boldsymbol{v}_u$, user's preferences that have been changed due to the domain influence $\boldsymbol{v}_u^d$, and domain-specific confounders DSC. We have established an implicit relationship in *Eq.* (5). However, it is difficult to be understood by involving generalized recommendation method $f(\boldsymbol{v}_u, \boldsymbol{v}_j)$ and evaluation measure $\delta_{u,i}(y,\hat{y})$. Here, we present a more intuitive analysis with matrix factorization. Suppose $f(\boldsymbol{v}_u, \boldsymbol{v}_j) = \boldsymbol{v}_u^\top \cdot \boldsymbol{v}_j$ and $\delta_{u,i}(y,\hat{y}) = (y_{u,i} - \hat{y}_{u,i})^2$. Then *Eq.* (5) can be written as:

$$\frac{(\boldsymbol{v}_u^{d\top} \cdot \boldsymbol{v}_j - y)^2}{p_{ujd}} = (\boldsymbol{v}_u^\top \cdot \boldsymbol{v}_j - y)^2, \tag{13}$$

Fix the user u and domain d, Eq. (13) should hold for all item attribute j. So differentiate $\boldsymbol{v}_j$ on both sides, we obtain:

$$\boldsymbol{v}_u^d = \pm\sqrt{p_{ujd}} \cdot \boldsymbol{v}_u. \tag{14}$$

The above equation gives an intuitive relationship between user's general preferences $\boldsymbol{v}_u$ and domain-specific preference $\boldsymbol{v}_u^d$. $\sqrt{p_{ujd}}$ represents the extent to which users are affected in domain d and change their preferences towards item attributes j. The sign in the equation represents the direction of preferences user have changed. Furthermore, $\sqrt{p_{ujd}}$ is responsible for eliminating biases produced by domains from user's general preferences. And v_u plays an important role in transferring information across domains by Eq. (9).

Complexity Analysis. The time complexity for optimizing the parameters through IPS estimator depends on the recommendation method we choose. For example, the time complexity for FM is $O(kJ^2 \cdot |O|)$, where k is the dimension of user and item attribute embeddings, O is the set of observed data. The time complexities for the estimator restriction, domain level restriction and individual restriction can be roughly estimated as $O(|O| \cdot J)$, $O(U \cdot D)$ and $O(U \cdot D)$. In practice, D and J is far less than $|O|$, so that the time complexity is similar to the typical latent factors models. The model parameters include user embeddings, item attribute embeddings and propensity scores. The size of propensity scores P is $U \cdot J \cdot D$. In total, the size of model parameters is linear with the input size and is close to the size of typical latent factors models. Indeed, we do not need to train the parameters of $f(v_u^d, v_j^d)$ in each domain serially through IPS estimator, so that we can train Θ_f in parallel.

5 EXPERIMENTS

In this section, we conduct experiments to validate the effectiveness of the proposed framework for cross-domain recommendation.

Table 1: Statistics of datasets after preprocessing.

Dataset	Domain	#user/overlapped	#item	#interaction
Amazon	CD	50,277/19,626	27,344	761,171
	Movie	97,973/42,048	32,689	1,975,835
	Book	830,854/46,226	276,899	17,570,584
Taobao	Gouwuche	12,588/7,882	78,996	115,152
	Sousuo	106,222/104,198	464,904	1,121,270
	Shoucai	107,589/104,225	559,796	1,786,916

5.1 Experimental Setup

5.1.1 Datasets. We use both *public* and *industry* datasets for evaluation.

• **Amazon Dataset** [34] contains product metadata from Amazon, including (user, item, rating) tuples and product metadata (category information, price, brand) from different domains. We perform a cross-domain rating prediction task on this dataset. We choose three relevant domains to test the debias effect of our method, namely, *CD* (named "CDs and Vinyl" in Amazon), *Movie* (named "Movies and TV" in Amazon) and *Book*.

• **Taobao Dataset** collects user behaviors from Taobao's recommender systems. In our experiment, we only use the click behaviors and the product metadata (category information, seller, brand and price). We perform a cross-domain click-through-rate prediction task on this dataset. Similarly, we choose three relevant applications as domains, namely, *Gouwuche*, *Sousuo* and *Shoucai*. In *Gouwuche*, users prefer recommendations based on their previously bought products. In *Sousuo*, users usually want to see the products that match their keywords better. In *Shoucai*, most users like to watch products, not necessarily for the purpose of buying something.

5.1.2 Evaluation Setting. In both datasets, we consider the product metadata as *item attributes*. Following previous studies [14, 22], we filter out the users with fewer than 10 interactions and the items with fewer than 30 interactions in each domain. The statistics of the datasets after preprocessing are shown in Table 1. In each domain, we take 50% of the interaction records as the training set, 30% as the validation set and 20% as the test set. Since we perform different tasks on different datasets, we set up different evaluation metrics for the two tasks. For rating prediction task, we adopt two widely used metrics, namely Mean Squared Error (MSE) and Mean Absolute Error (MAE). For click-through-rate prediction task, we adopt Area Under Curve (AUC) and Logloss following existing works [12].

5.1.3 Comparison Models. We adopt both single-domain and cross-domain baseline models for comparison:

• **BiasMF [24]** is a simple and well-known method for single-domain recommendation to improve the performance of SVD by introducing systematic biases associated with users and items.

• **FM [37]** is a powerful single-domain framework for collaborative filtering recommendation as an extension of a linear model designed to capture interactions between features.

• **Wide & Deep [9]** is a widely used single-domain recommendation method with jointly trained wide linear models and deep neural networks to combine memorization and generalization.

• **DeepFM [12]** is a deep learning based single-domain recommendation method which combines factorization machines for recommendation and deep learning for feature learning.

• **CMF [40]** is a multi-relation cross-domain learning approach which jointly factorizes matrices of individual domains by sharing user factors in different domains.

• **Multi-View DNN [10]** is a multi-learning framework for cross-domain user modeling in recommendation system, which maps two different views of the data into a shared view.

• **CoNet [16]** is the latest collaborative cross networks for cross-domain recommendation, which can enable dual knowledge transfer across domains by introducing cross connections.

Overall, our baselines have a good coverage of both single-domain and cross-domain baselines. Note that our framework can apply to various recommendation algorithms. In our experiments, we select FM, Wide & Deep and DeepFM for extension. To reproduce the experiments, we present detailed parameter configuration about baselines and our model in Appendix B.

5.2 Evaluation on Main Results

In this section, we perform the evaluation for cross-domain recommendation task on both public and industry dataset.

5.2.1 Evaluation with Public Dataset. We report the performance of different methods on rating prediction task with Amazon dataset in Table 2. From the results, it can be observed that:

(1) Among the four single-domain recommendation baselines (*i.e.,* BiasMF, FM, Wide & deep, DeepFM), BiasMF performs worst because it simply factorizes the user-item interaction, without modeling the item attributes. FM outperforms BiasMF by modeling context features. While Wide & Deep integrates linear models and deep neural networks to model the interactions, which performs

Table 2: Performance comparisons of different methods on public dataset. Smaller value is better.

Dataset	Category	Method	CD		Movie		Book	
			MSE	MAE	MSE	MAE	MSE	MAE
Amazon	Single domain	BiasMF	0.9907	0.7783	1.0618	0.7329	0.9584	0.7722
		FM	0.8563	0.6545	0.9728	0.7115	0.7958	0.6282
		Wide & deep	0.8487	0.6458	0.9687	0.7028	0.7846	0.6110
		DeepFM	0.8397	0.6397	0.9568	0.6914	0.7817	0.6096
	Cross domain	CMF	1.0949	0.7054	1.0796	0.6944	0.9283	0.6465
		Multi-View DNN	0.7958	0.6886	0.9437	0.6713	0.7636	0.6129
		CoNet	0.7813	0.6719	0.9319	0.6628	0.7637	0.5636
		FM+Our method	0.7741	0.6765	0.9263	0.6586	0.7581	0.5157
		Wide & deep+Our method	0.7618	0.6598	0.9127	0.6519	0.7490	0.5111
		DeepFM+Our method	**0.7590**	**0.6528**	**0.9079**	**0.6479**	**0.7416**	**0.5058**

Table 3: Performance comparisons of different methods on industry dataset. "↑" ("↓") indicates larger (smaller) is better.

Dataset	Category	Method	Gouwuche		Sousuo		Shoucai	
			AUC("↑")	LogLoss("↓")	AUC("↑")	LogLoss("↓")	AUC("↑")	LogLoss("↓")
Taobao	Single domain	BiasMF	0.6081	0.603	0.6381	0.572	0.6418	0.539
		FM	0.6336	0.536	0.6943	0.519	0.6950	0.496
		Wide & deep	0.6516	0.519	0.7149	0.490	0.7012	0.485
		DeepFM	0.6579	0.518	0.7194	0.487	0.7164	0.480
	Cross domain	CMF	0.6123	0.587	0.6450	0.553	0.6651	0.512
		Multi-View DNN	0.7317	0.521	0.7328	0.469	0.6835	0.472
		CoNet	0.7375	0.491	0.7839	0.429	0.7258	0.463
		FM+Our method	0.7182	0.475	0.8005	0.397	0.7187	0.497
		Wide & deep+Our method	0.7414	0.439	0.8119	0.389	0.7298	0.423
		DeepFM+Our method	**0.7424**	**0.412**	**0.8135**	**0.364**	**0.7316**	**0.421**

better. DeepFM achieves the best performance because it specially models neural feature interaction.

(2) Among the cross-domain recommendation baselines, CMF performs worse than some single-domain methods because it does not model the interactions between features. The other two baselines outperform all the single-domain baselines. However, they only mitigate the bias in domains by fine-tuning in the target domain, without accurately determining the confounding factors contained in the migrated information.

(3) Finally, it is obvious that our proposed framework performs consistently better than all of the single-domain and cross-domain baselines under different metrics. In particular, it largely improves the three base models (*i.e.,* FM, Wide&Deep and DeepFM). These results show that our framework is effective to improve the performance of cross-domain recommendation, and it is general to apply various recommendation algorithms. Such advantages are brought by the fact that we design a novel IPS estimator and learn the propensity scores by three restrictions to debias the confounder factors in domains.

5.2.2 Evaluation with Industry Dataset. As introduced in Section 5.1.1, we conduct the evaluation with three industry datasets, namely *Gouwuche*, *Sousuo* and *Shoucai*. The three datasets were collected from the logs of deployed recommender systems from Mobile Taobao and Tmall platforms. The main results comparing our proposed method with baselines are presented in Table 3. Overall, the experiment results are similar to the experiments on Amazon dataset, which further demonstrates the effectiveness of our framework. Additionally, the performances on the Sousuo dataset have improved more obviously, possibly because there are more overlapped users on the Sousuo dataset. Our method improves the most, which also shows that debiasing learning in our method is effective.

5.3 Method Analysis

In this section, we present more experiments and detailed analysis of the proposed method.

5.3.1 Ablation Study. The major novelty of our proposed framework is that it utilizes a novel IPS estimator and learns the propensity scores with three restrictions, namely estimator-level restriction, domain-level restriction and individual-level restriction. To examine the contributions of these three restrictions, we test the performance of three variants of the proposed framework based on DeepFM by removing each restriction from the full framework. Specifically, we consider the following variants for ablation study:

• *w/o estimator-level restriction*: In this variant, we remove the estimator-level restriction in Eq. (8).

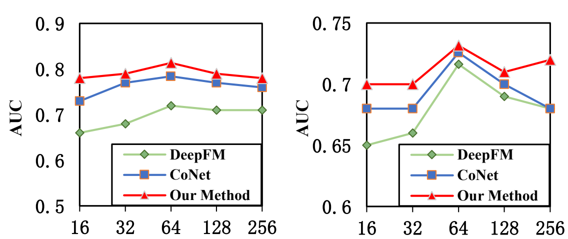
• *w/o domain-level restriction*: In this variant, we remove the domain-level restriction in Eq. (10).

• *w/o individual-level restriction*: In this variant, we remove the individua-level restriction in Eq. (11).

Table 4: Ablation study of the proposed method in terms of AUC. The best performances are marked bold on TAOBAO.

Method	Gouwuche	Sousuo	Shoucai
Full	**0.7424**	**0.8135**	**0.7316**
w/o estimator-level	0.6815	0.7527	0.6834
w/o domain-level	0.7178	0.7736	0.6996
w/o individual-level	0.7327	0.7973	0.7251

We report the experimental results of our full model and these variants in Table 4. As we can see, the performance of our full model is better than all of these variants, which indicates that the three restrictions are important for the final recommendation performance. In particular, the performance of the variant without estimator restriction is the worst one, showing that the relation between user’s general preference and domain effect is important for the propensity score learning.



(a) Sousuo dataset. (b) Shoucai dataset.

Figure 3: Performance of different embedding sizes (k).

5.3.2 Parameter Sensitivity Study. We further investigate the influence of model parameters on the performance to verify its robustness. We choose the best single-domain recommendation baseline and the best cross-domain recommendation baseline for comparison. Due to the space limit, we report the results on Sousuo dataset in Figure 3(a) and the results on Shoucai dataset in Figure 3(b). These two figures present the results for tuning the size of user and item attribute embeddings. We vary the dimension k in a set $\{16, 32, 64, 128, 256\}$. It can be seen that the performance improves with the increase of k and reaches the peak when $k = 64$. The performance of our model is consistently better than other baselines.

5.3.3 Evaluation on Overlapped Users. Intuitively, cross-domain recommendation mainly helps improve the recommendation for *overlapped users* who appear in multiple domains, since we can derive a more comprehensive user preference with transferred user information across domains. However, it is not clear whether a cross-domain approach would improve the recommendation for *non-overlapped users*. Since our test sets contains both overlapped and non-overlapped users, we further compare the performance on different user sets. From Table 5, we can see that our method outperforms the baselines on both overlapped and non-overlapped user sets. A major reason is that our framework explicitly models the attributes in mitigating the domain biases, so that it can also leverage cross-domain information to improve the recommendation performance for non-overlapped users.

Table 5: Evaluation on different user sets in terms of AUC.

Domain	User set	DeepFM	CoNet	Our method
Gouwuche	Overlap	0.6619	0.7563	0.7619
	Unoverlap	0.6449	0.6904	0.7248
Sousuo	Overlap	0.7268	0.8036	0.8317
	Unoverlap	0.7037	0.7538	0.8018
Shoucai	Overlap	0.7218	0.7402	0.7497
	Unoverlap	0.7136	0.7098	0.7246

5.3.4 Case Study. To better understand the learned general and domain-specific preference, we present a case study in Table 6. We present one sampled user and one sampled item. Both the user and item are randomly selected from our Taobao dataset. We apply our framework to the user-item pair. For this user, we learn the general and domain-specific preference. For the item, we learn the attribute embeddings. Next, we compute the cosine similarity between general (or domain-specific) preference and attribute embeddings in each domain. We further normalize the values between 0 and 1 with min-max normalization. As shown in Table 6, the results with general preference show that this user mainly likes the item on *category* and *brand*. Furthermore, in Sousuo or Gouwuche, purchase-guided domain setting is the major influencing factor, resulting in a high match between users and categories. While in Shoucai, exploration-guided domain setting is the major influence factor, leading to the shift of user preferences to brand and price. These results show that our framework is effective to mitigate the domain biases, and better capture general and domain-specific preference.

Table 6: Attribute importance comparison for a sampled user-item pair in three domains (FC = First Category, SC = Second Category). The best two attributes are marked bold.

	FC	SC	Brand	Seller	Price
General Preference	0.65	**0.81**	**0.72**	0.63	0.36
Sousuo	**0.74**	**0.78**	0.65	0.43	0.12
Gouwuche	**0.62**	**0.71**	0.56	0.29	0.18
Shoucai	0.64	0.73	**0.86**	0.65	**0.78**

6 CONCLUSIONS

In this paper, we presented a debiasing learning based cross-domain recommendation framework with causal embedding. Our proposed framework contained a novel IPS estimator and three restrictions for propensity score learning, namely restriction with estimator, restriction in domain level and restriction in individual level. Our framework is able to eliminate the biases produced in domains and transfer the debiased user information to other domains for recommendation. Extensive experiments on both public dataset and industry dataset have demonstrated the superiority of the proposed framework. Currently, we only deal with the selection bias produced in different domains. As future work, we will consider how to mitigate other biases like position bias and exposure bias.

ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China under Grant No. 61872369 and 61832017, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China under Grant No. 18XNLG22 and 19XNQ047, and Beijing Academy of Artificial Intelligence (BAAI). This work was also partially supported by Alibaba Group through Alibaba Innovative Research Program. Xin Zhao is the corresponding author.

REFERENCES

[1] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided exposure bias in recommendation. *arXiv preprint arXiv:2006.15772* (2020).

[2] Susan Athey, Guido W Imbens, and Stefan Wager. 2016. Approximate residual balancing: De-biased inference of average treatment effects in high dimensions. *arXiv preprint arXiv:1604.07125* (2016).

[3] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-domain mediation in collaborative filtering. In *ICUM*. Springer, 355–359.

[4] Da Cao, Xiangnan He, Liqiang Nie, Xiaochi Wei, Xia Hu, Shunxiang Wu, and Tat-Seng Chua. 2017. Cross-platform app recommendation by jointly modeling ratings and texts. *TOIS* 35, 4 (2017), 1–27.

[5] Kwun Chuen Gary Chan, Sheung Chi Phillip Yam, and Zheng Zhang. 2016. Globally efficient non-parametric inference of average treatment effects by empirical balancing calibration weighting. *Journal of the Royal Statistical Society. Series B, Statistical methodology* 78, 3 (2016), 673.

[6] Allison JB Chaney, David M Blei, and Tina Eliassi-Rad. 2015. A probabilistic model for using social networks in personalized item recommendation. In *RecSys*. 43–50.

[7] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *arXiv preprint arXiv:2010.03240* (2020).

[8] Jiawei Chen, Can Wang, Martin Ester, Qihao Shi, Yan Feng, and Chun Chen. 2018. Social recommendation with missing not at random data. In *ICDM*. IEEE, 29–38.

[9] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[10] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*. 278–288.

[11] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-domain Recommendation Without Sharing User-relevant Data. In *WWW*. 491–502.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Jens Hainmueller. 2012. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political analysis* (2012), 25–46.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[15] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *ICML*. PMLR, 1512–1520.

[16] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *CIKM*. 667–676.

[17] Kosuke Imai and Marc Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society: Series B: Statistical Methodology* (2014).

[18] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

[19] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

[20] Meng Jiang, Peng Cui, Nicholas Jing Yuan, Xing Xie, and Shiqiang Yang. 2016. Little is much: Bridging cross-platform behaviors through overlapped crowds. In *AAAI*.

[21] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *TOIS* 25, 2 (2007), 7–es.

[22] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *CIKM*. 1563–1572.

[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

[25] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems. In *WWW*. 817–818.

[26] Tzu-Heng Lin, Chen Gao, and Yong Li. 2019. Cross: Cross-platform recommendation for social e-commerce. In *SIGIR*. 515–524.

[27] Roderick JA Little and Donald B Rubin. 2019. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons.

[28] Yiming Liu, Xuezhi Cao, and Yong Yu. 2016. Are You Influenced by Others When Rating? Improve Rating Prediction by Conformity Modeling. In *RecSys*. 269–272.

[29] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *ECIR*. Springer, 656–661.

[30] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *SIGIR*. 203–210.

[31] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π-Net: A Parallel Information-sharing Network for Shared-account Cross-domain Sequential Recommendations. In *SIGIR*. 685–694.

[32] Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2012. Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv:1206.5267* (2012).

[33] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *RecSys*. 5–12.

[34] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*. 188–197.

[35] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for selection bias in learning-to-rank systems. In *Proceedings of The Web Conference 2020*. 1863–1873.

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

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

[38] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.

[39] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*. PMLR, 1670–1679.

[40] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *SIGKDD*. 650–658.

[41] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *SIGKDD*. 713–722.

[42] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: Discerning multi-faceted trust in a connected world. In *WSDM*. 93–102.

[43] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced negative sampling over knowledge graph for recommendation. In *Proceedings of The Web Conference 2020*. 99–109.

[44] Huan Yan, Chunfeng Yang, Donghan Yu, Yong Li, Depeng Jin, and Dah-Ming Chiu. 2019. Multi-site user behavior modeling and its application in video recommendation. *TKDE* (2019).

[45] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*. 279–287.

[46] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin. 2017. Selection of negative samples for one-class matrix factorization. In *ICDM*. SIAM, 363–371.

[47] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. *arXiv preprint arXiv:2011.01731* (2020).

[48] Ning Zhou and Jun-Yuan Xie. 2011. Select web services based on qualitative multi-users preferences. *Dianzi Xuebao(Acta Electronica Sinica)* 39, 4 (2011).

APPENDIX

We offer some reproducibility-related information as supplementary materials to help authors reproduce our method. The appendix includes 1 page and is organized into two sections:

- Additional details for our method are presented in Appendix A; and
- Additional details for our experiments are presented in Appendix B.

A ADDITIONAL DETAILS FOR OUR METHOD

A.1 Algorithm and Training

The parameters to learn in our model include the parameters for the recommendation function in $f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d)$ denoted by Θ_f and debiasing parameters $\{p_{ujd}, \boldsymbol{W}_1, \boldsymbol{w}_2, \boldsymbol{W}_3, b_1, b_2\}$ denoted as Θ_d. We learn Θ_f by the proposed Inverse-Propensity-Scoring Estimator in *Eq.* (3). The propensity score and parameters in *Eq.* (7) are learned by integrating the three loss terms in *Eq.* (8), *Eq.* (10) and *Eq.* (11):

$$L = Loss_1 + Loss_2 + Loss_L. \tag{15}$$

In each iteration, we first train the Θ_f in each domain and then train the Θ_d across domains. The whole method is trained with back-propagation. The training algorithm is summarized in Algorithm 1.

Algorithm 1 The training algorithm for our proposed method.

Initialize parameters.
while not convergence **do**
  for $d \in \mathcal{D}$ **do**
    Compute $\hat{R}_{IPS}(\hat{Y}|P)$;
    Update parameters in f with optimizer;
    $\boldsymbol{v}_{u,d} = \boldsymbol{W}_3(\boldsymbol{W}_1\boldsymbol{v}_u^d + \boldsymbol{w}_2 p_{ujd} + b1) + b2$;
    $e_1^d = \frac{\delta(f(\boldsymbol{v}_u^d, \boldsymbol{v}_j^d), y)}{p_{ujd}}$;
    $e_2^d = \delta(f(\boldsymbol{v}_{u,d}, \boldsymbol{v}_j^d), y)$;
  end for
  $\boldsymbol{v}_u^* = 1/D \sum_{d=1}^{D} \boldsymbol{v}_{u,d}$;
  $Loss = Loss_1 + Loss_2 + Loss_L$;
  Update parameters $p_{ujd}, \boldsymbol{W}_1, \boldsymbol{w}_2, \boldsymbol{W}_3, b_1, b_2$ with optimizer.
end while

A.2 Inference Details

In our framework, the number of dimensions for user embeddings and item attribute embeddings is set to 64. The values of λ_d and α_d are set to the multiplicative inverse of the number of samples in corresponding domain. We use Adam optimizer [23] with learning rate of 0.001 for optimization. The code used by our experiments is implemented with PyTorch in Python 3.6.

B ADDITIONAL DETAILS FOR EXPERIMENTS

B.1 Baseline Implementation

Table 7 represents the parameter settings of different methods used in our experiments.

Baselines	Settings
BiasMF	Embed.-size=64, batch-size=1024, Adam optimizer, learning-rate=0.001
FM	Embed.-size=64, batch-size=1024, Adam optimizer, learning-rate=0.0001
Wide & Deep	Embed.-size=64, batch-size=1024, Adam optimizer, dropout=0.5, hidden-size=256, 128, 64, learning-rate=0.0001
DeepFM	Embed.-size=64, batch-size=1024, Adam optimizer, dropout=0.5 hidden-size=256, 128, 64, learning-rate=0.0001
CMF	Embed.-size=64, batch-size=1024, Adam optimizer, learning-rate=0.0001
Multi-View DNN	Embed.-size=64, batch-size=1024, Adam optimizer, dropout=0.5 hidden-size=256, 128, 64,32, learning-rate=0.0001
CoNet	Embed.-size=64, batch-size=1024, Adam optimizer, dropout=0.5 hidden-size=256, 128, 64, learning-rate=0.0001

Table 7: Parameter settings of different baselines.