Interpolative Distillation for Unifying Biased and Debiased Recommendation

Sihao Ding
dsihao@mail.ustc.edu.cn
University of Science and Technology of China

Fuli Feng*
fulifeng93@gmail.com
University of Science and Technology of China, CCCD Key Lab of MCT

Xiangnan He
xiangnanhe@gmail.com
University of Science and Technology of China, CCCD Key Lab of MCT

Jinqiu Jin
jjq20021015@mail.ustc.edu.cn
University of Science and Technology of China

Wenjie Wang
wenjiewang96@gmail.com
National University of Singapore

Yong Liao
yliao@ustc.edu.cn
University of Science and Technology of China, CCCD Key Lab of MCT

Yongdong Zhang
zhyd73@ustc.edu.cn
University of Science and Technology of China

ABSTRACT

Most recommender systems evaluate model performance offline through either: 1) *normal biased test* on factual interactions; or 2) *debiased test* with records from the randomized controlled trial. In fact, both tests only reflect part of the whole picture: factual interactions are collected from the recommendation policy, fitting them better implies benefiting the platform with higher click or conversion rate; in contrast, debiased test eliminates system-induced biases and thus is more reflective of user true preference. Nevertheless, we find that existing models exhibit trade-off on the two tests, and there lacks methods that perform well on both tests.

In this work, we aim to develop a win-win recommendation method that is strong on both tests. It is non-trivial, since it requires to learn a model that can make accurate prediction in both factual environment (*i.e.,* normal biased test) and counterfactual environment (*i.e.,* debiased test). Towards the goal, we perform environment-aware recommendation modeling by considering both environments. In particular, we propose an *Interpolative Distillation* (InterD) framework, which interpolates the biased and debiased models at user-item pair level by distilling a student model. We conduct experiments on three real-world datasets with both tests. Empirical results justify the rationality and effectiveness of InterD, which stands out on both tests especially demonstrates remarkable gains on less popular items.

CCS CONCEPTS

• **Information systems → Recommender systems**.

*Corresponding author: Fuli Feng (fulifeng93@gmail.com).

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532002

KEYWORDS

Recommendation, System-induced Biases, Debiasing, Distillation

ACM Reference Format:
Sihao Ding, Fuli Feng, Xiangnan He, Jinqiu Jin, Wenjie Wang, Yong Liao, and Yongdong Zhang. 2022. Interpolative Distillation for Unifying Biased and Debiased Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532002

1 INTRODUCTION

Recommender system is a medium to connect users and content, playing an increasingly important role in Web applications such as e-commerce [44], social media [9], and content sharing [42]. Offline model performance evaluation is indispensable in recommender system, which can be divided into two categories.

- *Normal biased test* [16, 21, 28], which evaluates on the interactions collected from factual environment with recommendation policy. Models performing well under this test are supposed to benefit the platform *w.r.t.* click or conversion rate.
- *Debiased test* [7, 26, 36], which evaluates on interactions collected from a randomized controlled trial (*i.e.,* random exposure). Since system-induced biases [27] (*e.g.,* exposure bias and popularity bias) are eliminated in this counterfactual environment, this kind of evaluation can better reflect user preference.

Considering that each evaluation protocol emphasizes the benefit of one side, it is desirable to have models that perform well on both tests. If a model is strong on both offline tests rather than one, the platform will be more comfortable to launch it for real use.

Nevertheless, existing work usually adopts either test, leading to two types of recommendation models: biased and debiased. *Biased models* [15, 16] are directly trained from historical interactions in the factual environment. However, they could leverage the bias as the shortcut for model fitting, over-recommending popular items and neglecting user preference on non-mainstream items [1, 25]. *Debiased models* eliminate the data biases with revised training and/or inference, *e.g.,* inverse propensity scoring (IPS) [7, 36], causal

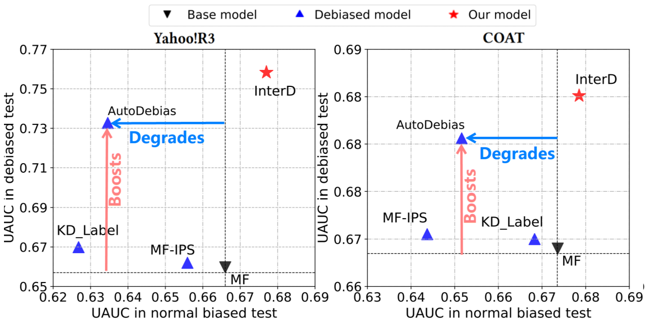


Figure 1: Performance trade-off of existing methods on *normal biased test* and *debiased test* on Yahoo!R3 and Coat. For instance, taking MF as the base model, AutoDebias boosts in *unbiased test* but degrades in *normal biased test*.

representation learning [26, 51], causal adjustment [44, 49], etc. Nevertheless, it is difficult to control the strength of debiasing. For example, some representative work [3, 38, 44, 51] pursues an even item distribution[1] in the recommendation lists, which has the risk of over-debiasing and hurting the click/conversion rate [49, 50, 52].

We find that debiasing typically results in performance trade-off between the two tests. Figure 1 provides an empirical evidence where we adopt Matrix Factorization (MF) [21] as the base model and apply three representative debias strategies KD_Label [26], IPS [36] and AutoDebias [7]. As can be seen, although the debiased models improve MF on *debiased test*, their performance on *normal biased test* is reduced significantly. Since the two tests measure different aspects of the system, we desire to have models that are strong in both tests. This requires to learn an environment-aware model that can make accurate prediction in both environments.

Towards this goal, an intuitive solution is unifying the data from both environments to train a model. However, random exposure is at the high expense of hurting user experience, so the data collected from the counterfactual environment is usually much smaller [7, 26, 38]. As such, the data from counterfactual environment can hardly play a role by a simple data merging [4, 7, 29]. Another direct solution is to train the biased and debiased model separately and then ensemble them. However, it is unclear the two models are strong/weak at which types of users or items, and existing ensemble strategies [12] are not tailored for this win-win recommendation scenario. To tackle these challenges, we need to aggregate the biased and debiased models with fine-grained coefficients at the level of user-item pair.

In this work, we propose an *Interpolative Distillation* (InterD) framework that treats the biased and debiased models as teachers for label-based distillation. The InterD distills knowledge of two environments from the teachers to a student model, and automatically adjusts the coefficient for each teacher at the user-item pair level. Our consideration for assigning the coefficient is that the student should trust the more reliable teacher on each user-item pair. We thus set the coefficient based on the distance between the teacher's prediction and the rating for observed user-item pairs. We also consider incorporating unobserved data (*i.e.,* missing data) into InterD, and set the coefficient of unobserved pairs as the distance between the student's prediction and the teacher's prediction. We believe that starting with learning easier aspects of knowledge can facilitate knowledge absorption of the student [6, 46], which means relying more on the closer teacher. Empirical results show that InterD can outperform both biased and debiased models on both tests, especially on the less popular items, validating the rationality and effectiveness of InterD.

To summarize, our main contributions are as follows:

- Revealing the trade-off issue of existing recommendation methods and formulate a new win-win recommendation problem.
- Solving the problem through environment-aware modeling and a new *Interpolative Distillation* framework.
- Conducting extensive experiments to justify the rationality and effectiveness of InterD.

2 RELATED WORK

Debiased Recommendation. Recommender systems utilize user feedback (*e.g.,* rating and click) for model learning while user feedback has intrinsic biases [8], such as selection bias [38], conformity bias [51], popularity bias [49], and position bias. To mitigate the biases, many solutions have been proposed, including *data imputation* [40, 43], *regularization* [3], and *causal inference* [33]. Specifically, data imputation estimates the effect of missing data by training an additional imputation model [43] to alleviate selection bias. Regularization can mitigate the biases in the recommendation lists by introducing a regularizer into the training and/or inference [10]. Causal inference studies on debiasing usually follow two classic causal frameworks: *potential outcome* [35] and *structural causal models* [33]. In particular, IPS [38] and *doubly robust (DR) learning* [43] are two popular methods in the potential outcome framework, which are applied to reduce various bias issues, such as popularity bias [7] and selection bias [43]. AutoDebias [7], promotes existing IPS and DR methods by optimizing parameters for debiasing with meta-learning, achieving state-of-the-art debiased performance. For structural causal models, most researchers retrospect the data generation process and reduce the bias through causal intervention [41, 49] or counterfactual inference [42, 44]. Despite their great success, previous works pursue higher performances on the debiased test but ignore the recommendation accuracy on the normal biased test, which may hurt the benefits of the platform. It is thus risky for the platform to adopt such methods without biased test. Our work pursues win-win recommendation to achieve superior performance on both biased and debiased tests.

Distillation for Recommendation. In the past several years, witnessing the great success of neural network techniques, more complicated neural network approaches are applied in recommender systems [16, 48]. Although the cumbersome neural network improves the accuracy of the recommendation, it also brings high computational complexity and large storage requirements. To overcome these shortcomings, a line of research employs a knowledge distillation mechanism to distill a small student model from the teacher [14, 23, 24]. For example, RD [39] is a ranking distillation framework to distill a student model by learning from both the data and the teacher model. DE-RRD [20] distills the latent knowledge and relaxed ranking information of teacher model to a student

[1]As items have different quality and attractiveness, pursuing an even distribution of recommended items is not optimal.

model. Our work is different from exiting distillation works since we aim to leverage the knowledge of two environments from teacher models to develop a win-win recommender model rather than pruning for cost-cutting purposes. To ensure the fairness of performance comparison, we keep the student model has the same space cost (*i.e.,* embedding size) as the teacher model in our work.

Multistakeholder Recommendation. Recommender systems have multiple stakeholders, including the user, item providers, and the platform [2, 31, 34]. In the light that focusing on the benefit of user is unfair to the other stakeholders, multistakeholder recommendation accounts for more stakeholders. For example, FairRec [32] reveals the two user groups of the two-sides platform as multistakeholder, such as Uber drivers and passengers. They map the multistakeholder recommendation problem to the fairly allocating indivisible goods problem, and alleviating the trade-off of stakeholders. TFROM [45] treats different item providers as multistakeholder, and regulates the recommendation result to be fair across different item providers while serving users. Beyond the fairness across users or items, our work considers the benefits of the platform and users.

3 TASK FORMULATION

Under the probabilistic view, the goal of recommendation can be seen as estimating the probability distribution $P(R|U,I)$ [37], which indicates the rating (R) likelihood between user (U) and item (I). With the assumption that R follows a Gaussian distribution [37], we can train a recommender model $f(;\theta)$ as the expectation of $P(R|U,I)$, *i.e.,* the mean of the Gaussian distribution. After training on historical data, the model scores each user-item pair by $\hat{r} = f(u,i;\theta)$ and generates personalized ranking accordingly.

Let $E \in \{e_b, e_d\}$ denotes the environment variable to describe which environment an interaction from: e_b for factual environment (with bias) and e_d for counterfactual environment (without bias). $\mathcal{D}_b = \{(u,i,r)|u \in \mathcal{U}, i \in \mathcal{I}\}$ denotes the biased interactions, which are collected from the previous recommendation policy in the factual environment. r denotes the labeled rating value[2], and $\mathcal{U}$ and $\mathcal{I}$ denote the user set and item set, respectively. Similarly, $\mathcal{D}_d = \{(u,i,r)|u \in \mathcal{U}, i \in \mathcal{I}\}$ denotes the interactions collected from a randomized controlled trial (RCT) that is free from the impact of system-induced biases. There is no overlap with $\mathcal{D}_b$ and $\mathcal{D}_d$. Note that $|\mathcal{D}_d|$ is typically much smaller than $|\mathcal{D}_b|$ since random exposure is costly and hurts user experience [7, 26, 38].

In this work, we aim to build a model that performs well on both *normal biased test* and *debiased test*. Let $\mathcal{T}_b$ and $\mathcal{T}_d$ be the holdout testing data from e_b and e_d, respectively. We formulate the win-win recommendation problem as modeling $P(R|U,I)$ *s.t.* its expectation model $f(u,i;\theta)$ can perform well on both $\mathcal{T}_b$ and $\mathcal{T}_d$.

4 METHODOLOGY

In this section, we conduct environment-aware modeling of $P(R|U,I)$, and then elaborate the proposed *Interpolative Distillation* for model learning.

[2]In this paper, we follow [7] to use binary ratings as $r \in \{-1, 1\}$ which can be easily generalized to multiple levels.

4.1 Environment-aware Modeling

A straightforward method to estimate $P(R|U,I)$ for two tests is to directly train a model over historical interactions of two environments *i.e.,* $\mathcal{D}_b \cup \mathcal{D}_d$. However, the data of counterfactual and factual environments is highly imbalanced (*i.e.,* $|\mathcal{D}_b| \gg |\mathcal{D}_d|$), making the utility of $\mathcal{D}_d$ be overwhelmed by $\mathcal{D}_b$. Thus, we decompose $P(R|U,I)$ to make it environment-aware:

$$P(R|U,I) = \frac{P(R,U,I)}{P(U,I)} \tag{1a}$$

$$= \frac{\sum_E P(R|U,I,E)P(U,I,E)}{P(U,I)} \tag{1b}$$

$$= \frac{\sum_E P(R|U,I,E)P(E|U,I)P(U,I)}{P(U,I)} \tag{1c}$$

$$= \sum_E P(R|U,I,E)P(E|U,I). \tag{1d}$$

Specifically, Eq. (1a) follows the Bayes' theorem; Eq. (1b) follows the law of total probability; Eq. (1c) also adopts Bayes' theorem over $P(U,I,E)$. Since $E \in \{e_b, e_d\}$, according to Eq. (1d), we can estimate $P(R|U,I)$ by separately modeling the underlying probability distributions: $P(R|U,I,E=e_b)$, $P(R|U,I,E=e_d)$, and $P(E|U,I)$.

- $P(R|U,I,E=e_b)$ denotes the rating distribution in the factual environment with bias, *i.e.,* items are exposed according to the deployed recommendation policy.
- $P(R|U,I,E=e_d)$ represents the rating distribution in the counterfactual environment, where system-induced biases are blocked, *i.e.,* imagining that items are randomly exposed to user under a randomized controlled trial (RCT).
- $P(E|U,I)$ is the posterior distribution of environment given a user-item pair: $P(E=e_b|U,I)$ and $P(E=e_d|U,I)$ reflect how likely the pair belongs to the factual and counterfactual environments, respectively. $P(E=e_b|U,I) > P(E=e_d|U,I)$ implies that the rating of the user-item pair is more likely from the factual environment. That is, $P(E|U,I)$ implies how likely the rating of a pair is affected by system-induced biases. For instance, *recommendation takers* who tend to follow the recommender system have higher $P(E=e_b|U,I)$ than *recommendation ignorers* [13], since ratings made by *recommendation takers* are highly likely affected by biases.

According to Eq. (1d), we can explain why the existing models encounter a trade-off on the two tests — they estimate $P(R|U,I)$ under only one environment. Consequently, biased and debiased models make recommendation with $P(R|U,I,E=e_b)$ and $P(R|U,I,E=e_d)$, respectively. As such, they only achieve strong performance under either the *normal biased test* or *debiased test*. On the contrary, we consider the whole picture as a mixture of both environments, aiming to achieve strong performances on both tests.

After decomposing $P(R|U,I)$ to be environment-aware, we can separately estimate $P(R|U,I,E=e_b)$, $P(R|U,I,E=e_d)$ and $P(E|U,I)$ to uncover $P(R|U,I)$. As the first two distributions can be estimated by biased and debiased models[3], respectively. The key to estimate $P(R|U,I)$ lies in a proper modeling of $P(E|U,I)$, which means given

[3]We refer this survey [8] for the detail of debiased training methods, which mainly focus on accounting for both $\mathcal{D}_b$ and $\mathcal{D}_d$ to obtain more accurate estimation of $P(R|U,I,E=e_d)$ than learning directly from $\mathcal{D}_d$.

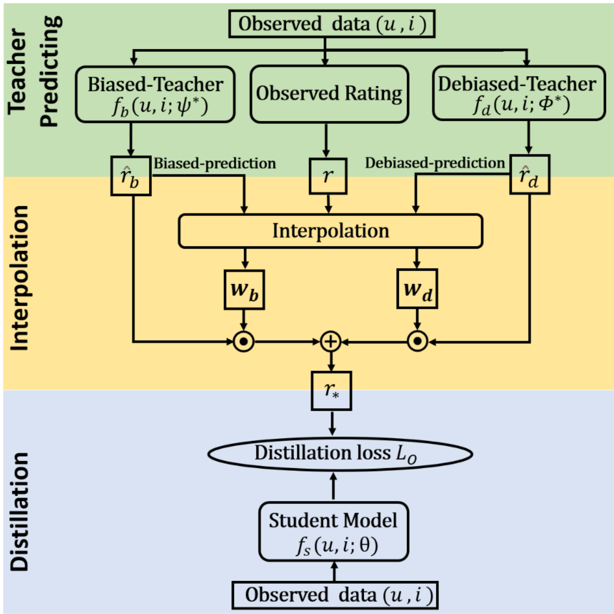


Figure 2: The procedure of InterD. From top to bottom, 1) fetch predictions from teachers for each observed user-item pair (u, i); 2) calculate interpolation coefficients (*i.e.,* w_b and w_d) and r_* for (u, i); 3) train student model with distillation loss L_O. The figure only illustrates InterD with observed data (u, i). As to unobserved data, InterD replaces r with student model prediction $\hat{r}$ to generate r'_* and calculates imputation distillation loss L_N.

a user-item pair, estimating how likely it belongs to each environment. Nevertheless, directly estimating $P(E|U, I)$ from $\mathcal{D}_b \cup \mathcal{D}_d$ (*e.g.,* learning a binary classifier) also suffers from high data imbalance (*i.e.,* $|\mathcal{D}_b| \gg |\mathcal{D}_d|$). The classifier will be over confident on the major class $E = e_b$.

4.2 Interpolative Distillation

We next consider how to uncover the environment-aware $P(R|U, I)$ by estimating: ① $P(R|U, I, E)$ and ② $P(E|U, I)$ as in Eq. (1d).

4.2.1 Estimate $P(R|U, I, E)$. Note that $P(R|U, I, E = e_b)$ and $P(R|U, I, E = e_d)$ represent the rating distributions conditioned on a specific environment e_b or e_d. Furthermore, they imply two different rating distributions when the corresponding pair is affected or not affected by bias. As there has been extensive work on normal biased [21] and debiased models [7], we directly use them to train a biased model $f_b(;\psi^*)$ and a debiased model $f_d(;\phi^*)$ for $P(R|U, I, E = e_b)$ and $P(R|U, I, E = e_d)$ respectively, where ψ^* and ϕ^* represent the optimized model parameters. Assuming $P(R|U, I, E = e_b)$ and $P(R|U, I, E = e_d)$ follow the Gaussian distribution (as the tradition in [37]), the model outputs $\hat{r}_b = f_b(u, i; \psi^*)$ and $\hat{r}_d = f_d(u, i; \phi^*)$ are the estimated expectations of $P(R|U, I, E = e_b)$ and $P(R|U, I, E = e_d)$, respectively. And we term $\hat{r}_b$ and $\hat{r}_d$ as biased-prediction and debiased-prediction, respectively.

4.2.2 Estimate $P(E|U, I)$. For notation briefness, given a user-item pair (u, i), we denote the environment posterior distributions $P(E = e_b|U, I)$ and $P(E = e_d|U, I)$ as w_b and w_d, respectively. As they interpolate the two rating distributions, we term them as **interpolation coefficients** [5]. Similar as [30], we assign w_b and w_d according to the loss of biased and debiased model predictions $\hat{r}_b$ and $\hat{r}_d$. Formally,

$$w_b = \frac{L_b(\hat{r}_b, r)^{\gamma_1}}{L_b(\hat{r}_b, r)^{\gamma_1} + L_d(\hat{r}_d, r)^{\gamma_1}}, w_d = \frac{L_d(\hat{r}_d, r)^{\gamma_1}}{L_b(\hat{r}_b, r)^{\gamma_1} + L_d(\hat{r}_d, r)^{\gamma_1}}, \tag{2}$$

where r denotes the observed rating for the (u, i) pair, $L_b(\cdot)$ denotes a normal recommendation loss such as mean squared error (MSE); $L_d(\cdot)$ denotes a debiased loss function such as an IPS weighted MSE; and γ_1 is a **negative** hyper-parameter that can smooth the interpolation coefficients. Increasing the absolute value of γ_1 will increase the weight of the model that has the smaller loss.

The intuition behind Eq. (2) is that, if the given (u, i) pair more likely belongs to one environment, the corresponding prediction $\hat{r}_b$ or $\hat{r}_d$ should be closer to the observed rating r. Taking the counterfactual environment as an example, the debiased-prediction $\hat{r}_d$ is the estimated expectation of $P(R|U, I, E = e_d)$, closer to the expectation means higher probability that the observed rating r comes from $P(R|U, I, E = e_d)$, $P(E = e_d|U, I)$ should thus be larger. And we use the loss value $L_d(\hat{r}_d, r)$ to measure the distance between $\hat{r}_d$ and r, since $w_d \propto L_d(\hat{r}_d, r)^{\gamma_1}$, the negative value of γ_1 ensures that low loss $L_d(\hat{r}_d, r)$ leads to large w_d. As the same, w_b expresses how likely the rating of a (u, i) pair will be observed in the factual environment (*i.e.,* $E = e_b$), which is inversely proportional to the distance between biased-prediction and observed rating (*i.e.,* $L_b(\hat{r}_b, r)$). Finally, w_b and w_d are normalized by the same denominators.

4.2.3 Distillation. According to Eq. (1) we have:

$$\begin{aligned} P(R|U, I) &= P(E = e_b|U, I) \cdot P(R|U, I, E = e_b) \\ &+ P(E = e_d|U, I) \cdot P(R|U, I, E = e_d). \end{aligned} \tag{3}$$

We estimate $P(E = e_d|U, I)$ and $P(E = e_d|U, I)$ by w_b and w_d respectively, according to Eq. (2). And in Sec. 4.2.1 we have estimated the expectations of $P(R|U, I, E = e_b)$ and $P(R|U, I, E = e_b)$ as $\hat{r}_b = f_b(u, i; \psi^*)$ and $\hat{r}_d = f_d(u, i; \phi^*)$, respectively. Since $P(R|U, I)$ follows a Gaussian distributions, we can estimate its expectation by interpolating $\hat{r}_b$ and $\hat{r}_d$. Formally, for each observed (u, i) pair in $\mathcal{D}_b \cup \mathcal{D}_d$, we have:

$$r_* = w_b\hat{r}_b + w_d\hat{r}_d, \tag{4}$$

where r_* denotes the expectation of $P(R|U, I)$, which can perform well on both tests. We next consider how to obtain r_* for testing user-item pairs. Note that we cannot directly use r_* for ranking non-interacted items, since calculating it requires the observed rating r. We thus employ a label-based distillation mechanism to distill a model $f_s(u, i; \theta)$ with the r_* on observed data. The distilled student model $f_s(u, i; \theta)$ can generate the predictions that follow $P(R|U, I)$ to perform well on both tests.

Formally, we optimize a distillation loss,

$$L_O = \frac{1}{|\mathcal{D}_b| + |\mathcal{D}_d|} \sum_{(u,i,r) \in \mathcal{D}_b \cup \mathcal{D}_d} L\left(f_s(u, i; \theta), r_*\right). \tag{5}$$

Note that we omit the L_2 regularization for briefness. As $f_s(u, i; \theta)$ is learned from the interpolation of biased and debiased models, we term this training procedure (Figure 2) as *Interpolative Distillation*, naming $f_s(u,i;\theta)$ as the *student model*; $f_b(u,i;\psi^*)$ and $f_d(u,i;\phi^*)$ are *biased-teacher* and *debiased-teacher*, respectively. The student model $f_s(u,i;\theta)$ adopts the same model configuration as teacher models to keep the representation ability. InterD is thus different from conventional distillation methods that aim to reduce model size for saving costs.

4.2.4 Incorporate unobserved data. Most recommendation data is highly sparse, *i.e.,* only a small portion of user-item ratings are observed (*cf.* Table 1). The distillation objective in Eq. (5) ignores the missing data, which is known to be useful for item recommendation [17]. In this light, we then consider how to incorporate the unobserved data into the InterD framework. One belief is that the biased and debiased models have encoded the knowledge of the two environments. Accounting for their predictions on the unobserved pairs has the potential of enhancing the student model.

Let $\mathcal{D}_n = \mathcal{U} \times \mathcal{I} - \mathcal{D}_b \cup \mathcal{D}_d$ denote the unobserved data, where $\mathcal{U} \times \mathcal{I}$ denotes the whole set of user-item pairs. To account for unobserved data, the key lies in generating r'_* for (u,i) in $\mathcal{D}_n$. Similar to Eq. (4), we define r'_* as:

$$r'_* = w'_b \hat{r}_b + w'_d \hat{r}_d, \tag{6}$$

where w'_b and w'_d are the interpolation coefficient of unobserved pairs corresponding to $P(E=e_b|U,I)$ and $P(E=e_d|U,I)$, respectively. Similar to Eq. (2), we define w'_b and w'_d as:

$$w'_b = \frac{L_b(\hat{r}_b,\hat{r})^{\gamma_2}}{L_b(\hat{r}_b,\hat{r})^{\gamma_2} + L_d(\hat{r}_d,\hat{r})^{\gamma_2}},\ w'_d = \frac{L_d(\hat{r}_d,\hat{r})^{\gamma_2}}{L_b(\hat{r}_b,\hat{r})^{\gamma_2} + L_d(\hat{r}_d,\hat{r})^{\gamma_2}}, \tag{7}$$

where γ_2 is also a negative hyper-parameter; and $\hat{r}$ is the output of student model $f_s(u,i;\theta)$. Based on the r'_*, we further define an *imputation distillation loss* L_N over the unobserved data $\mathcal{D}_n$, which is formulated as:

$$L_N = \frac{1}{|\mathcal{D}_n|} \sum_{(u,i)\in\mathcal{D}_n} L\left(f_s(u,i;\theta), r'_*\right). \tag{8}$$

By combining L_O and L_N, we obtain the final distillation objective with consideration of both observed and unobserved data. Formally, we optimize the student model $f_s(\cdot;\theta)$ to estimate the expectation of $P(R|U,I)$ by minimizing:

$$\theta^* = \arg\min_{\theta}\ (L_O + \beta L_N), \tag{9}$$

where β is a non-negative hyper-parameter to adjust the contribution of the *imputation distillation loss* L_N.

Note that $L_b(\hat{r}_b,\hat{r})$ and $L_d(\hat{r}_d,\hat{r})$ in Eq. (7) denote the distances between predictions of teachers and the student. As γ_2 is negative, a larger distance leads to a smaller interpolation coefficient. It means the student model will learn more from the closer teacher over unobserved data. In other words, the student tends to learn the easier aspects of knowledge since the smaller distance makes it easier to follow the corresponding teacher. This is similar to curriculum learning [6, 46]. Notably, r'_* in L_N will be updated once the student model has been updated during the distillation, since the calculation of w'_b and w'_d relies on the student prediction. This procedure is consistent with self-paced learning [19, 22].

Another interpretation of Eq. (7) is that the student prediction $\hat{r}$ can be seen as an imputation value [11, 43] over the unobserved pair. As the distillation proceeds, the student model continuously learns from the two teachers over the labeled pairs through L_O. The student will gradually accumulate the knowledge about $P(R|U,I)$ and generate the more accurate imputation value. We thus postulate that the distance between $\hat{r}$ and $\hat{r}_b$ (or $\hat{r}_d$) can reflect how likely an unobserved pair belongs to the factual (or counterfactual) environment and implies how reliable the biased-teacher (or debiased-teacher) is (*cf.* Table 4).

Algorithm 1: Interpolative Distillation

```
Input: Biased data D_b and unbiased data D_d.
Output: A win-win recommender model f_s(·; θ*).
Train a biased model f_b(·; φ) as biased-teacher;
Train a debiased model f_d(·; ψ) as debiased-teacher;
Initialize the student model f_s(u, i; θ);
while Stop condition is not reached do
    Fetch (u, i) pairs from U × I ;
    if (u, i) ∈ D_b ∪ D_d then
        Calculate w_d and w_b with Eq. (2);
        Generate r_* with Eq. (4);
    else
        Calculate w'_d and w'_b with Eq. (7);
        Generate r'_* with Eq. (6);
    end
    Update f_s(·; θ) with Eq. (9);
end
Return the student model f_s(·; θ*);
```

To summarize, compared with conventional recommender model training, our InterD demonstrates three main differences:

- InterD leverages both biased model and debiased model instead of choosing only one specification.
- InterD distills a student model from the biased and debiased models to handle both factual and counterfactual environments.
- InterD accounts for the unlabeled user-item pairs when distilling the student model.

Lastly, we apply InterD over MF [21] and AutoDebias [7], and elaborate its detailed procedure in Algorithm 1.

5 EXPERIMENTS

We conduct experiments to answer the following questions:

- **RQ1:** Does our proposed InterD outperform biased and debiased models on the two tests?
- **RQ2:** Why does InterD perform well on both tests?
- **RQ3:** What factors influence the effectiveness of InterD?

5.1 Experimental Settings

Datasets. To validate the effectiveness of InterD, we utilize three datasets with RCT data in different application domains: 1) Yahoo!R3[4], 2) Coat[5], and 3) Product[6], which are obtained from the music, coat, and micro-video recommendation services, respectively. All datasets contain both normal biased data $\mathcal{D}_b$ collected from

[4]https://webscope.sandbox.yahoo.com/.
[5]https://www.cs.cornell.edu/~schnabts/mnar/.
[6]It is a popular micro-video sharing platform.

Table 1: Statistics of the datasets, NB-Tr and RCT-Tr are short for normal biased training data and RCT training data, respectively. NB-Te and RCT-Te are short for normal biased testing data and RCT testing data, respectively.

Dataset	#User	#Item	#NB-Tr	#RCT-Tr	#Val	#NB-Te	#RCT-Te
Yahoo!R3	15.4k	1.0k	249k	5.4k	33.8k	31.2k	48.6k
Coat	290	300	5.6k	464	928	696	4.1k
Product	7.1k	10.7k	1,060k	27k	146k	132.5k	243k

historical interactions and the RCT data $\mathcal{D}_d$ acquired by a random exposure policy [38]. Following [7], we partition the RCT data into the RCT training data **(5%)**, RCT validation data **(5%)**, and RCT testing data **(90%)**. Additionally, we extract **10%** normal biased data for *normal biased test* and treat the remaining 90% data as biased training set. In this work, we combine the RCT training data and normal biased training data to optimize InterD, and all baselines leverage the RCT validation data to choose hyper-parameters. As to Yahoo!R3 and Coat, explicit feedback with ratings larger than 3 is treated as 1, otherwise the feedback is labeled as -1. For the Product dataset, the ratings are based on user's playing time, which are defined by the platform according to its business logic.

Compared methods. We compare InterD with its base model MF, debiased models from advanced debias strategies, and model learned with multistakeholder objectives. In particular,

- **MF [21]:** it is a widely used benchmark model in recommendation. We train MF with the combination of normal biased and RCT training data, following the setting of MF (combine) in [7]. Note that incorporating RCT training data can enhance the performance of MF on both tests.
- **MF-IPS [38]:** we apply the classic IPS method to MF, where the calculation of propensity scores follows [7, 38].
- **AutoDebias [7]:** it is a SOTA debiased model trained with normal biased and RCT training data. It optimizes propensity scores and an imputation model over RCT training data. We adopt the source code and hyper-parameter ranges for grid search released in the original paper.
- **MF-PD [49]:** we enhance MF with the Popularity-bias Deconfounding (PD) [49] technique, which leverages causal intervention to reduce bias but focuses on popularity bias. We search the hyper-parameter γ that controls the smoothness of popularity effect in {1e-6, 5e-6, $\cdots$, 5e-1}.
- **KD_Label [26]:** it is a SOTA debias knowledge distillation method that distills the knowledge of debiasing from a teacher model to a student model. We train the teacher model with RCT data and optimize the student model with the teacher outputs and biased training data. We adopt the public implementation and tune the hyper-parameters following the original paper.
- **MF-TFROM [45]:** TFROM is one of SOTA multistakeholder recommendation methods, we apply it to MF and treat different item groups as stakeholder according to item popularity. We belief that controlling across groups will adjust the effect of system-induced biases, such as popularity bias.
- **Ensemble** directly fuses the predictions of MF and AutoDebias linearly via constant coefficients during the inference stage, which also leverages the knowledge of two teachers. We search the coefficient as the hyper-parameters with the step of 0.1.
- **InterD**[7] takes MF and AutoDebias as biased-teacher and debiased-teacher, and uses MSE in the distillation objective to optimize the student model. Te student model is another MF that has the same embedding size as biased-teacher and debiased-teacher.

Evaluation Metrics. We adopt *UAUC* and *NDCG@K* [18] to evaluate the recommendation performance. For each user, we calculate the AUC [7] and NDCG@K over the exposed items in the testing data, and then take the average scores of all users[8] to obtain UAUC and NDCG@K, respectively. Here K is set as 5 for all datasets, and we omit K for simplicity in the following sections. Besides, we adopt two metrics for clearly measuring the overall performance on the two tests: *F1-UAUC* and *F1-NDCG*, which are calculated by the harmonic mean of the scores on the debiased test and normal biased test. In particular,

$$\textit{F1-UAUC} = \frac{2 \times UAUC_{DT} \times UAUC_{NBT}}{UAUC_{DT} + UAUC_{NBT}},$$
$$\textit{F1-NDCG} = \frac{2 \times NDCG_{DT} \times NDCG_{NBT}}{NDCG_{DT} + NDCG_{NBT}}, \tag{10}$$

where UAUC_{DT} and UAUC_{NBT} denote the UAUC scores on the debiased test and normal biased tests, respectively. NDCG_{DT} and NDCG_{NBT} denote the NDCG scores on the debiased test and normal biased test, respectively.

5.2 Performance Comparison (RQ1)

Table 2 reports the performance comparison under two tests on three datasets. From the table, we have the following observations:

- Compared with the base model MF, all debiased model perform better on the *debiased test* but show inferior performance on the *normal biased test*. In other words, there is a clear trade-off between the two tests, *i.e.,* debias methods promote the debiased test performance with significant sacrifice of the biased test performance. We postulate the reason is that these methods estimates the rating distribution $P(R|U, I)$ as $P(R|U, I, E = e_b \text{ or } e_d)$, ignoring the counterpart. To pursue a win-win recommendation, it is thus reasonable to perform environment-aware modeling.
- InterD achieves the best overall performance regarding both F1-UAUC and F1-NDCG across the three datasets. It demonstrates the effectiveness of InterD in tackling the performance trade-off on the two tests. We attribute the performance gain to the proposed distillation objective, which considers both observed and unobserved data, and combines the knowledge of the two environments at fine-grained level of user-item pairs.
- **Remarkably, InterD outperforms all baselines regarding debiased test performance on all datasets.** For instance, on the Product dataset, InterD achieves relative performance improvements of 3.5% and 12.56% *w.r.t.* UAUC and NDCG over the second-best AutoDebias on debiased test. These results indicate that InterD can achieve the SOTA debias performance. One reason of the performance gain is InterD can alleviate the over-debias issue of AutoDebias (more details in Sec. 5.3). Another possible reason is that InterD optimizes model parameters with

[7] The code of proposed method InterD is available at https://github.com/Dingseewhole/InterD_master

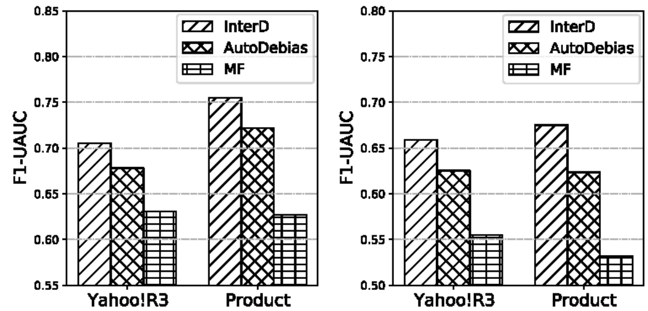
[8] We remove the testing users that only have positive (*i.e.,* 1) or only have negative (*i.e.,* -1) testing samples, since their NDCG and AUC are always equal to 1 or 0.

Table 2: Recommendation performances on Yahoo!R3, Coat, and Product. The best and second best results are highlighted with bold and underline, respectively. DT and NBT are short for debiased test and normal biased test. "*" denotes the best performance is significantly better than all baselines based on paired t-test at the significance level of 0.05.

Yahoo!R3	DT		NBT		Overall	
	UAUC	NDCG	UAUC	NDCG	F1-UAUC	F1-NDCG
MF	0.6597	0.5545	0.6660	0.8310	0.6628	0.6651
MF-IPS	0.6606	0.5552	0.6559	0.8250	0.6583	0.6637
KD_Label	0.6699	0.5760	0.6268	0.8128	0.6477	0.6742
AutoDebias	0.7327	0.6441	0.6346	0.8168	0.6802	0.7202
MF-PD	0.7232	0.6397	0.6647	0.8290	0.6927	0.7221
MF-TFROM	0.6602	0.5554	0.6658	0.8304	0.6630	0.6656
Ensemble	0.7460	0.6570	0.6614	0.8292	0.7012	0.7331
InterD	**0.7583**	**0.6764**	**0.6770**	**0.8388**	**0.7153***	**0.7489***
Coat	UAUC	NDCG	UAUC	NDCG	F1-UAUC	F1-NDCG
MF	0.6690	0.4941	0.6736	0.8277	0.6713	0.6188
MF-IPS	0.6705	0.5081	0.6437	0.8037	0.6568	0.6090
KD_Label	0.6780	0.5059	0.6306	0.8038	0.6534	0.6210
AutoDebias	0.6806	0.5268	0.6516	0.8204	0.6658	0.6416
MF-PD	0.6710	0.5252	0.5756	0.7940	0.6197	0.6322
MF-TFROM	0.6701	0.5021	0.6669	0.8228	0.6685	0.6236
Ensemble	0.6822	0.5232	0.6431	0.8158	0.6621	0.6375
InterD	**0.6851**	**0.5270**	**0.6785**	**0.8295**	**0.6818***	**0.6445***
Product	UAUC	NDCG	UAUC	NDCG	F1-UAUC	F1-NDCG
MF	0.5965	0.1344	**0.7443**	**0.4956**	0.6623	0.2114
MF-IPS	0.7219	0.1395	0.6712	0.4342	0.6956	0.2111
KD_Label	0.7380	0.1460	0.6306	0.4262	0.6801	0.2175
AutoDebias	0.8473	0.3210	0.6670	0.4263	0.7464	0.3662
MF-PD	0.7658	0.1467	0.6541	0.4279	0.7056	0.2185
MF-TFROM	0.7034	0.1375	0.6954	0.4443	0.6994	0.2100
Ensemble	0.8274	0.3185	0.6901	0.4410	0.7525	0.3699
InterD	**0.8773**	**0.3615**	0.7206	0.4732	**0.7913***	**0.4099***

all user-item pairs, which tackles some optimization issues of AutoDebias and KD_label caused by learning debias parameters only with RCT training data, which is usually very small.

- Ensemble also performs better than some debias methods under debiased tests in some cases such as the Yahoo!R3 dataset. This result indicates that jointly leveraging the outputs of biased and debiased teachers can counterweight some issues of the teachers. This result is consistent with previous studies on multi-teacher aggregation [47]. Moreover, it justifies the rationality of decomposing $P(R|U,I)$ and separately estimating the underlying distributions, instead of directly learning $P(R|U,I)$ from the training data.
- In particular, InterD consistently surpasses Ensemble under both tests on all datasets, while they have the same teachers. Noticing that Ensemble use constant coefficients across all user-item pairs, we attribute the performance gain of InterD to the fine-grained user-item level interpolation coefficients. In other words, inferring the environment posterior $P(E|U,I)$ from model prediction distances is reasonable, leading to a more accurate interpolative distribution of $P(R|U,I)$.
- AutoDebias steadily outperforms MF-PD, KD_Label and MF-TFROM under debiased test, which shows that leveraging the meta-learning mechanism to learn debias parameters with RCT



(a) 80% less popular items

(b) 30% unpopular items

Figure 3: Performances regarding the most 80% and 30% unpopular items on Yahoo!R3 and Product datasets.

data can still help debias. This is consistent with the findings in the original paper of AutoDebias [7].

- In comparison with other debias methods, MF-TFROM fails to demonstrate superior performance on *debiased test*, but it mostly sacrifices minimal *normal biased test* performance. It confirms that multiple objectives can alleviate the performance trade-off issue. Nevertheless, it is unlikely to outperform biased and debiased models on their strong test.

5.3 Exploratory Analysis (RQ2).

Performances on Less Popular Items. To further justify the superiority of InterD, we test it on the less popular items of testing data. The popularity of an item is determined by its frequency in the training data. Figure 3(a) and Figure 3(b) show the F1-UAUC regarding 80% and 30% most unpopular items, respectively. We omit this result on Coat since it only has 300 unique items. According to the figures, InterD consistently outperforms its teachers on the recommendations of less popular items. Furthermore, InterD achieves higher gains on F1-UAUC of the 30% unpopular items as compared with the 80% unpopular items. These two findings demonstrate InterD can promote teachers recommendation performances on less popular items (more empirical evidence are shown in Figure 5(a)(c)).

Study on the Interpolation Coefficient. To study how interpolation coefficients (*i.e.,* w_d and w_b) promote InterD, we visualize the changes of interpolation coefficients as the popularity of items increases in Yahoo!R3. We visualize the results on Yahoo!R3 since it is public and relatively larger. Figure 4 shows that as the item popularity increases, the value of w_d keeps decreasing while w_b keeps enlarging. The trend indicates that, in the distillation phase, InterD trusts the debiased-teacher more on less popular items, but relies more on the biased-teacher for popular items. This is reasonable since during the training phase of teachers, the debiased-teacher generates larger inverse propensity scores for less popular items [7, 36], making the model pay more attention to these items. Thus the debiased-teacher learns better representations of less popular items than the biased-teacher that without debias strategy. On the contrary, the biased-teacher naturally focuses more on popular items due to their massive interactions in the training data, which produces better representations of popular items. Recall that we

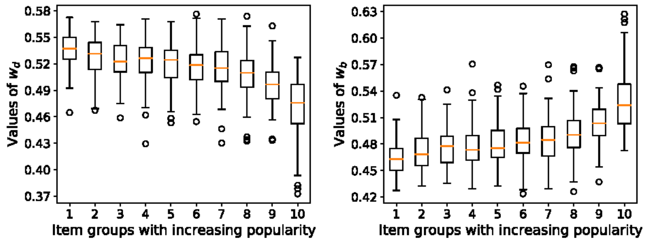
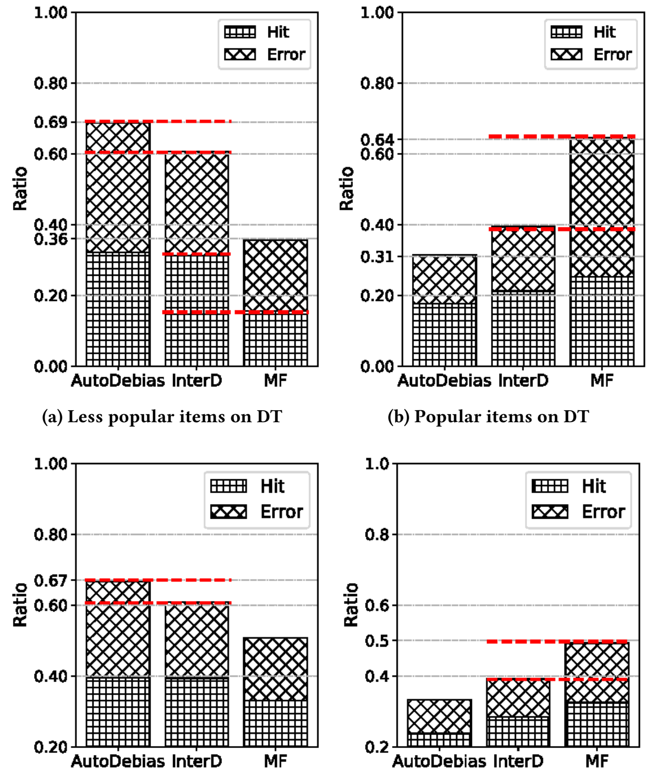


Figure 4: The sensitivity of interpolation coefficients with increasing item popularity in Yahoo!R3.

design InterD for finely aggregating the knowledge of both environments, Figure 4 proves that InterD can fine-grained adjust the interpolation coefficient according to the advantages of teachers.

Recommendation Comparison *w.r.t.* Item Popularity. Table 2 shows that InterD achieves good performance on both tests, and mostly beats its teachers in their advantageous test (*e.g.,* beats MF on NBT meanwhile beats AutoDebias on DT on Yahoo!R3). To investigate the underlying reasons, we compare the recommendation results of InterD and its two teachers AutoDebias and MF. Specifically, 1) we collect top-5 items for each user recommended by AutoDebias, MF, and InterD. 2) Then we split all items in top-5 recommendation results into the popular set (Figure 5(b)(d)) and less popular set (Figure 5(a)(c)) based on item popularity in training data. We group the top 20% most popular items into the popular set, while the other items are in the less popular set. 3) We separately visualize the results on DT (Figure 5(a)(b)) and NBT (Figure 5(c)(d)). From the figures, we have the following observations:

- By comparing Figure 5(a) and Figure 5(b), we find that in recommendation results of AutoDebias, 69% items belong to less popular set and 31% items belong to popular set. In contrast, only 36% of items recommended by MF are less popular, while 64% are popular. This finding proves that AutoDebias tend to recommend less popular items but MF tend to recommend popular items. The same conclusion can be drawn by comparing Figure 5(c) and Figure 5(d).
- We can find out why InterD always beats its debiased-teacher on DT by studying Figure 5(a). As compared to AutoDebias, InterD recommends fewer less-popular items, and the reduced recommendations are all from the error part (*i.e.,* negative testing samples). It means although InterD recommends fewer less-popular items, it maintains the same hit rate as AutoDebias, which improves accuracy. The study also confirms that AutoDebias has the over-debias issue that indiscriminately recommends less popular items, and InterD can alleviate the issue by fitting $r*$ that generated by two teachers.
- In Figure 5(a), compared with MF, InterD dramatically increases the hit rate, while keeping the same error rate of less popular items recommendations. And in Figure 5(b) InterD makes the almost same hit rate as MF, but greatly reduces the error rate. These observations demonstrate InterD can alleviate the problem about over-recommend popular items to achieve extremely better performance than MF in DT.
- Recall in Table 2, InterD mostly beats its biased-teacher MF on NBT. We can find out the reason in Figure 5(d), that InterD reduces the error rate of popular item recommendations.



(a) Less popular items on DT
(b) Popular items on DT
(c) Less popular items on NBT
(d) Popular items on NBT

Figure 5: The distribution of recommendation results on DT (a) and (b); and NBT (c) and (d) of Yahoo!R3. Hit and error denote the correct and incorrect recommendation results, respectively. For each method, the sum of popular item ratio and less popular ratio is 1.

- In Figure 5(a) and Figure 5(c), we can clearly see how InterD outperforms AutoDebias and MF for recommendations on less popular items. This study verifies that environment-aware recommendation modeling can improve the accuracy of recommendations for less popular items.

Case Studies. For further analysis, we conduct four case studies on Yahoo!R3. Table 3 shows four recommendation results, the former two results are collected from *normal biased test*, while the other two results are from *debiased test*. From Table 3 we can observe:

- For user-1747, MF recommends item-2 that is the 18th most popular item out of 1000 items. However, AutoDebias recommends very unpopular item-243 (614th/1000 popular) as top-1 recommendation. But the ground-truths of these two items are all negative (*i.e.,* user-1747 does not like item-2 and item-243), and only InterD makes the correct prediction of these items. From these observations we can find out MF and Autodebias are easily misled by popularity, resulting in bias amplification or over-debias issue. And it verifies only modeling the $P(R|U,I)$ under a specific environment is not good enough. Fortunately, AutoDebias makes the correct prediction on item-2, and MF makes the correct prediction on item-243. Thus InterD is able to counterbalance incorrect predictions of two teachers based on interpolative distillation.

Table 3: Case studies on Yahoo!R3. Pop rank denotes the popularity rank of an item *w.r.t.* its frequency in training data, the lower rank represents the higher popularity, pos/neg denote positive/negative *i.e.,* -1/1 rating, GT denotes the ground-truth of each interaction.

Test name	Statistics of user and item			Rank and recommend results			GT
	User	Item	Pop rank	MF	AutoDebias	InterD	
Normal	1747	2	18th/1000	top-1, pos	top-6, neg	top-5, neg	**neg**
bias test	1747	243	614th/1000	top-7, neg	top-1, pos	top-3, neg	**neg**
Debiased	684	788	709th/1000	top-6, neg	top-1, pos	top-5, neg	**neg**
test	684	37	15th/1000	top-1, pos	top-8, neg	top-8, neg	**neg**

- In *debiased test*, for user-684, Autodebias and MF also make the similar wrong recommendations: Autodebias recommends a very unpopular negative item-788 (709th/1000) as top-1 recommendation and MF recommends a very popular negative item-37 (15th/1000) as top-1 item. These incorrect recommendations are all corrected by InterD through aggregating the predictions of two teachers. It demonstrates InterD is a superior debiased model and makes examples to explain why InterD consistently outperforms debiased-teacher AutoDebias in *debiased test.*

5.4 In-depth Analyses (RQ3)

Ablation Study. In Table 4, we compare the InterD with InterD-B. The InterD-B is an Interpolative Distillation framework that calculates the interpolation coefficient with a binary method:

$$\begin{cases} w_d = 1, w_b = 0 & if \quad w_d > w_b \\ w_d = 0, w_b = 1 & if \quad w_d < w_b \end{cases}$$
$$\begin{cases} w'_d = 1, w'_b = 0 & if \quad w'_d > w'_b \\ w'_d = 0, w'_b = 1 & if \quad w'_d < w'_b \end{cases} \tag{11}$$

Obviously, in Table 4 the performance of InterD-B is always worse than InterD across all datasets on all metrics, it demonstrates that the fine-grained interpolative coefficient at the user-item pair level enhances the performance of InterD. And it also confirms the rationality of Eq. (2) and Eq. (7).

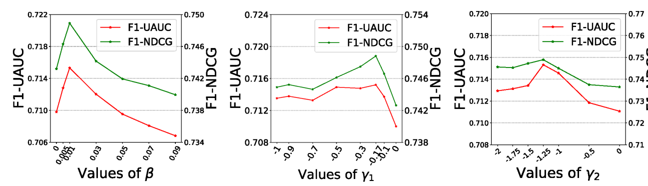
Another observation in Table 4 is the InterD consistently outperforms the InterD-O which does not leverage the unobserved data *i.e.,* setting $\beta \equiv 0$ in Eq. (9). It verifies our motivation of incorporating the unobserved data (*i.e.,* missing data) into InterD with imputation loss L_N to boost its performance.

Furthermore, we compare the InterD with InterD-I. The InterD-I is a variant of InterD, which adopts a default imputation method that sets the imputation values of all unobserved data as zero [11]. Thus, the InterD-I calculates w'_b and w'_d with replacing $\hat{r}$ by 0 in Eq. (7). It means InterD-I assumes that the teacher model whose prediction is closer to 0 is more reliable for unobserved data. Table 4 shows the InterD steadily outperforms InterD-I across all datasets. Since the student model accumulates the knowledge about $P(R|U,I)$ during distillation, its output can be viewed as the imputation value that is more accurate than the conventional imputation of unobserved data. These results also verify the rationality of Eq. (7).

Study of Hyper-parameters. We study how the hyper-parameters of InterD affect its performances on two tests. Figure 6(a) shows the performance of InterD as the value of β increasing. As can be seen, both F1-UAUC and F1-NDCG show a clear trend of increasing than decreasing. The increasing part reflects that incorporating unobserved data into InterD with relatively smaller value of β can enhance the student model. The decreasing part is also reasonable since too big β value will lead to more contributions of L_N, which dilutes the information of observed data from L_O. Figure 6(b) and Figure 6(c) show InterD is not particularly sensitive to γ_1 and γ_2, but when $\gamma_1 = 0$ or $\gamma_2 = 0$ InterD will degenerate into the variants of InterD-B, and the performance is worst. It proves that generating the interpolative coefficients by Eq. (2) and Eq. (7) is better than the binary method in Eq. (11).

Table 4: The performance of vanilla InterD, InterD-B, InterD-O and InterD-I in DT and NBT. DT and NBT are short of debiased test and normal biased test. The best results are highlighted with bold font.

Dataset	Method	DT		NBT		Overall	
		UAUC	NDCG	UAUC	NDCG	F1-UAUC	F1-NDCG
Yahoo!R3	InterD-B	0.6521	0.5433	0.6158	0.8048	0.6334	0.6487
	InterD-O	0.7521	0.6674	0.6720	0.8385	0.7098	0.7432
	InterD-I	0.7482	0.6657	0.6737	0.8373	0.7090	0.7417
	InterD	**0.7583**	**0.6764**	**0.6770**	**0.8388**	**0.7153**	**0.7489**
Coat	InterD-B	0.6696	0.4952	0.6713	0.8274	0.6704	0.6195
	InterD-O	0.6832	0.5184	0.6738	0.8288	0.6785	0.6378
	InterD-I	0.6849	0.5220	0.6762	0.8289	0.6805	0.6406
	InterD	**0.6851**	**0.5270**	**0.6785**	**0.8295**	**0.6818**	**0.6445**
Product	InterD-B	0.6840	0.2379	0.7012	0.4290	0.6925	0.3061
	InterD-O	0.8593	0.3242	0.6929	0.4513	0.7672	0.3773
	InterD-I	0.8671	0.3415	0.7014	0.4523	0.7755	0.3892
	InterD	**0.8773**	**0.3615**	**0.7206**	**0.4732**	**0.7913**	**0.4099**



(a) Study of β (b) Study of γ_1 (c) Study of γ_2

Figure 6: Performances of InterD as changing hyper-parameters β, γ_1 and γ_2 on Yahoo!R3.

6 CONCLUSION

In this work, we explored a new recommendation problem with both normal biased test in a factual environment (with bias); and debiased test in a counterfactual environment (without bias). We revealed that existing methods encounter trade-off between the two tests due to considering one specific environment and ignoring the other. To pursue win-win recommendation, we conducted environment-aware recommendation modeling with consideration of both environments. To tackle the estimation challenge we proposed the Interpolative Distillation (InterD) framework to fine-grained interpolate the rating distributions of the environments. We applied InterD on basic MF (biased-teacher) and AutoDebias (debiased-teacher), and conducted extensive experiments on three real-world datasets. Empirical results confirm InterD achieves the best performance on both tests in most cases. Besides, InterD achieves remarkable gains on less popular items.

This work opens up a new research direction in recommendation — developing win-win recommender systems that serve users better while benefiting platforms. In the future, we would like to extend InterD to more complex biased and debiased models to explore the performance ceiling of InterD. Moreover, we will test InterD under the setting of implicit feedback to validate InterD more comprehensively. Lastly, conducting theoretical analysis for InterD is also an interesting direction.

ACKNOWLEDGMENTS

This work is supported by National Key Research and Development Program under Grant 2021YFC3300500-02 and National Natural Science Foundation of China (U21B2026, 62121002) .

REFERENCES

[1] Himan Abdollahpouri. 2020. Popularity bias in recommendation: A multi-stakeholder perspective. *arXiv:2008.08551* (2020).

[2] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. 2020. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction* (2020), 127–158.

[3] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *RecSys*. 42–46.

[4] Ahmed M Alaa, Michael Weisz, and Mihaela Van Der Schaar. 2017. Deep counterfactual networks with propensity-dropout. *ICML*, 1–6.

[5] Yujia Bao, Shiyu Chang, and Regina Barzilay. 2021. Predict then Interpolate: A Simple Algorithm to Learn Stable Classifiers. In *ICML*. 640–650.

[6] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*. 41–48.

[7] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. In *SIGIR*. 21–30.

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

[9] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2019. Samwalker: Social recommendation with informative sampling strategy. In *WWW*. 228–239.

[10] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. Esam: Discriminative domain adaptation with non-displayed items to improve long-tail performance. In *SIGIR*. 579–588.

[11] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. 2006. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology* (2006), 1087–1091.

[12] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science* (2020), 241–258.

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

[14] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision* (2021), 1789–1819.

[15] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR*. 355–364.

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

[17] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*. 549–558.

[18] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* (2002), 422–446.

[19] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-paced curriculum learning. In *AAAI*. 2694–2700.

[20] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2020. DE-RRD: A Knowledge Distillation Framework for Recommender System. In *CIKM*. 605–614.

[21] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*. 426–434.

[22] M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-Paced Learning for Latent Variable Models. In *NIPS*. 1189–1197.

[23] Wonbin Kweon, SeongKu Kang, and Hwanjo Yu. 2021. Bidirectional Distillation for Top-K Recommender System. In *WWW*. 3861–3871.

[24] Jae-woong Lee, Minjin Choi, Jongwuk Lee, and Hyunjung Shim. 2019. Collaborative distillation for top-N recommendation. In *ICDM*. 369–378.

[25] Roger Zhe Li, Julián Urbano, and Alan Hanjalic. 2021. Leave No User Behind: Towards Improving the Utility of Recommender Systems for Non-mainstream Users. In *WSDM*. 103–111.

[26] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data. In *SIGIR*. 831–840.

[27] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. Mitigating Confounding Bias in Recommendation via Information Bottleneck. In *RecSys*. 351–360.

[28] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware message-passing gcn for recommendation. In *WWW*. 1296–1305.

[29] Min Lu, Saad Sadiq, Daniel J Feaster, and Hemant Ishwaran. 2018. Estimating individual treatment effect in observational data using random forest methods. *Journal of Computational and Graphical Statistics* (2018), 209–219.

[30] Yulei Niu and Hanwang Zhang. 2021. Introspective Distillation for Robust Question Answering. *NIPS*, 1–13.

[31] Sungkyu Park, Jamie Yejean Park, Hyojin Chin, Jeong-han Kang, and Meeyoung Cha. 2021. An Experimental Study to Understand User Experience and Perception Bias Occurred by Fact-checking Messages. In *WWW*. 2769–2780.

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

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

[34] Bashir Rastegarpanah, Krishna P Gummadi, and Mark Crovella. 2019. Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In *WSDM*. 231–239.

[35] Donald B Rubin. 2005. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association* (2005).

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

[37] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*. 1257–1264.

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

[39] Jiaxi Tang and Ke Wang. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *KDD*. 2289–2298.

[40] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural memory streaming recommender networks with adversarial training. In *KDD*. 2467–2475.

[41] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD*. 1717–1725.

[42] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*. 1288–1297.

[43] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *ICML*. 6638–6647.

[44] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *KDD*. 1791–1800.

[45] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. 2021. TFROM: A Two-sided Fairness-Aware Recommendation Model for Both Customers and Providers. In *SIGIR*. 1013–1022.

[46] Liuyu Xiang, Guiguang Ding, and Jungong Han. 2020. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *ECCV*. 247–263.

[47] Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. 2020. Model compression with two-stage multi-teacher knowledge distillation for web question answering system. In *WSDM*. 690–698.

[48] Yin Zhang, Derek Zhiyuan Cheng, Tiansheng Yao, Xinyang Yi, Lichan Hong, and Ed H. Chi. 2021. A Model of Two Tales: Dual Transfer Learning Framework for Improved Long-tail Item Recommendation. In *WWW*. 2220–2231.

[49] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR*. 11–20.

[50] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2021. Popularity Bias Is Not Always Evil: Disentangling Benign and Harmful Bias for Recommendation. *arXiv:2109.07946* (2021).

[51] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *WWW*. 2980–2991.

[52] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and mitigating item under-recommendation bias in personalized ranking systems. In *SIGIR*. 449–458.