Prior-Guided Accuracy-Bias Tradeoff Learning for CTR Prediction in Multimedia Recommendation

Dugang Liu
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University
Shenzhen, China
dugang.ldg@gmail.com

Yang Qiao
FiT, Tencent
Shenzhen, China
sunnyqiao@tencent.com

Xing Tang
FiT, Tencent
Shenzhen, China
xing.tang@hotmail.com

Liang Chen
FiT, Tencent
Shenzhen, China
leocchen@tencent.com

Xiuqiang He
FiT, Tencent
Shenzhen, China
xiuqianghe@tencent.com

Zhong Ming*
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University
Shenzhen, China
mingz@szu.edu.cn

ABSTRACT

Although debiasing in multimedia recommendation has shown promising results, most existing work relies on the ability of the model itself to fully disentangle the biased and unbiased information and considers arbitrarily removing all the biases. However, in many business scenarios, it is usually possible to extract a subset of features associated with the biases by means of expert knowledge, i.e., the confounding proxy features. Therefore, in this paper, we propose a novel debiasing framework with confounding proxy priors for the accuracy-bias tradeoff learning in the multimedia recommendation, or CP2Rec for short, in which these confounding proxy features driven by the expert experience are integrated into the model as prior knowledge corresponding to the biases. Specifically, guided by these priors, we use a bias disentangling module with some orthogonal constraints to force the model to avoid encoding biased information in the feature embeddings. We then introduce an auxiliary unbiased loss to synergize with the original biased loss in an accuracy-bias tradeoff module, aiming at recovering the beneficial bias information from the above-purified feature embeddings to achieve a more reasonable accuracy-bias tradeoff recommendation. Finally, we conduct extensive experiments on a public dataset and a product dataset to verify the effectiveness of CR2Rec. In addition, CR2Rec is also deployed on a large-scale financial multimedia recommendation platform in China and achieves a sustained performance gain.

*Corresponding author

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3613801

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Multimedia recommendation, Debiasing recommendation, Confounding proxy, Accuracy-bias tradeoff

ACM Reference Format:
Dugang Liu, Yang Qiao, Xing Tang, Liang Chen, Xiuqiang He, and Zhong Ming. 2023. Prior-Guided Accuracy-Bias Tradeoff Learning for CTR Prediction in Multimedia Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3613801

1 INTRODUCTION

Multimedia recommendation is a key component in many mobile Internet platforms, aiming to accurately recommend multimodal items to different users that they are more likely to be interested in [3, 24]. Click-through rate (CTR) prediction is an important technique to achieve this goal, but it is susceptible to various biases in user-system interactions, including the user-induced biases and the system-induced biases caused by the user characteristics and the deployed recommendation policies, respectively [5]. Since intervening with the users to control the user-induced biases may bring some potential risks, most existing works focus on how to effectively mitigate the system-induced biases.

Existing works for debiased recommendation can be mainly divided into two categories according to whether an additional subset of unbiased feedback is introduced or not, i.e., debiased learning with and without an unbiased feedback subset [2, 4, 14, 17, 28, 36], where this subset is collected by using a uniform policy instead of the recommendation policy to avoid as much as possible the source of system-induced biases. With the unbiased knowledge provided by a subset of unbiased feedback, the former aims to design various effective knowledge transfer strategies to guide the ideal training of recommendation models. When unbiased knowledge is not available, the latter introduces various techniques to ensure the unbiasedness of the optimization objective based on some prior modeling assumptions about the biases. Since collecting an unbiased feedback subset usually requires a large cost in many practical applications, we focus on the latter in this paper. Although debiasing learning without an unbiased feedback subset has shown promising results, most of them expect the model to be able to accurately distinguish the biased and unbiased information from all the input features, and consider arbitrarily removing all the biases. This is easy to bring the learning burden and performance bottleneck of the model.

In particular, in many business scenarios, a subset of features associated with biased information can be defined based on expert knowledge, which is called the *confounding proxy features* in this paper. Intuitively, introducing the confounding proxy features as the prior knowledge during the modeling process is expected to reduce the learning burden of the model and more effectively guide the model explicitly to avoid encoding biased information. Furthermore, we argue that some beneficial biases that fit these properties should not be completely eliminated, since the decision-making behavior of most users is still likely to be bias-driven. As shown in Figure 1, taking the homepage recommendation of a large-scale online financial management platform in China as an example, we can define a set of confounding proxy features such as position index, fund tags, fund yield trend chart, and fund average return rate. Moreover, the latter two features can be considered as beneficial biases, since cost-effective funds will get more exposure and are justified for user experience. This motivates us to explore a new debiasing framework to efficiently integrate the confounding proxy priors and achieve a better accuracy-bias tradeoff for multimedia recommendation.

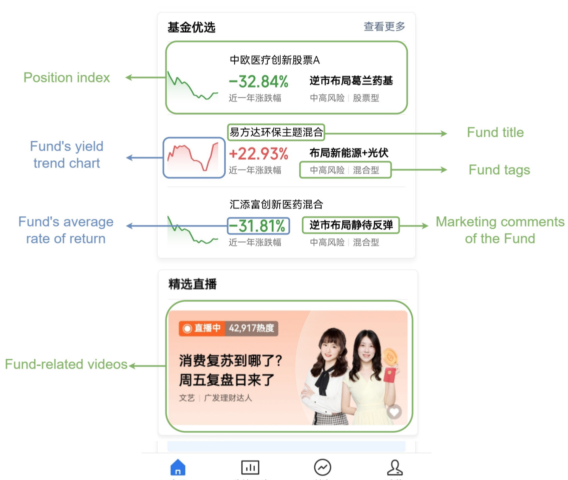


Figure 1: The homepage recommendation of a large-scale online financial management platform in China, including indications of various features, where the blue ones represent the features related to beneficial biases.

In this paper, we propose a novel accuracy-bias tradeoff recommendation framework via the confounding proxy priors, or CP2Rec for short. Specifically, our CP2Rec consists of three customized modules: 1) a feature segmentation module is used to screen a set of potential confounding proxy features from input features based on expert knowledge and serve as prior knowledge for model training; 2) a bias disentangling module is guided by these priors and leverages the orthogonality constraints between the representations to facilitate the model to remove the encoding of biased information from feature embeddings; and 3) an accuracy-bias tradeoff module aims to introduce an auxiliary unbiased loss to synergize with the original biased loss to extract the embeddings corresponding to the beneficial bias information from the embeddings of confounding proxies. They are then used in the inference stage together with the above-purified feature embeddings. Intuitively, we use the confounding proxy features in the second module to force the model to learn the bias-independent feature embeddings and recover the beneficial bias information in the feature embeddings in the third module, so as to achieve the accuracy-bias tradeoff. Finally, we conduct extensive offline and online experiments to verify the effectiveness of our CP2Rec.

2 RELATED WORK

In this section, we briefly review some relevant works on two research topics, including debiased learning with and without an unbiased feedback subset in recommender systems.

Debiased Learning with an Unbiased Feedback Subset. Compared with the possible risks of controlling the user-induced biases by intervening with the users, it is more feasible and secure to alleviate the system-induced biases by replacing the deployed recommendation policy [21]. By adopting a special uniform policy that does not rely on the recommendation policy for item delivery, but randomly selects and ranks the candidate set, a subset of unbiased feedback that can act as a good unbiased proxy can be collected [14, 16]. Therefore, existing work on this research topic aims at how to more effectively utilize a subset of unbiased feedback to guide an ideal model training and can be mainly divided into three routes, including inverse propensity score-based methods, multi-stage training-based methods, and joint training-based methods. An inverse propensity score-based method calculates the weight of each feedback through the above subsets and integrates it into the optimization objective for calibration of biased distributions [14, 15, 29, 34, 40]. A multi-stage training-based method utilizes some effective training frameworks that alternately use the biased and unbiased feedback to jointly learn a better set of unbiased parameters [4, 7, 11, 35]. A joint training-based method trains separate models for biased and unbiased feedback, respectively, and customizes some alignment strategies to directly constrain the joint learning of both [2, 14]. Different from these existing works, our CP2Rec does not rely on an unbiased feedback subset and expects to achieve an accuracy-bias tradeoff rather than arbitrarily removing all the biases.

Debiased Learning without an Unbiased Feedback Subset. Since collecting a subset of unbiased feedback will usually hurt the user experience and the platform revenue in some practical applications, the setting of this research topic is to consider debiasing learning when a subset of unbiased feedback is not available. Due to the lack of guidance from unbiased information, most of the existing work needs to make some prior modeling assumptions about the biases or introduce some specific techniques to ensure the unbiasedness of the optimization objective [9, 13, 19, 20, 22, 23, 38, 41]. They can be mainly divided into three routes, including inverse propensity score-based methods, ideal unbiased loss optimization-based methods, and disentangled representation-based methods. Note that unlike when an unbiased feedback subset is available, an inverse propensity score-based method now needs to estimate the weights based on some variables related to the users and items [2, 12]. An ideal unbiased loss optimization-based method aims to use some theoretical tools to derive a tractable unbiased loss or a generalization error upper bound for a specific biased problem so that it can be directly optimized [27, 28, 33, 37]. A disentangled representation-based method constrains the model from accurately identifying the biased and unbiased information from the input features and encoding them in the corresponding biased and unbiased components to alleviate the impact of biases on the obtained representations [17, 18, 36, 42]. Our CP2Rec falls into the last route. But different from existing works, we introduce the confounding proxy priors to ease the learning burden of the model, and further drive the accuracy-bias tradeoff to avoid some possible limitations in these works, such as removing some beneficial biases.

3 PRELIMINARIES

In this section, we formally define the accuracy-bias tradeoff recommendation task with the necessary notations when a subset of unbiased feedback is not available. A typical multimedia recommender system usually consists of a set of M users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, a set of N items $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, a set of J fields of user attributes $\mathcal{A} = \{\mathcal{A}^1, \mathcal{A}^2, \ldots, \mathcal{A}^J\}$ and a set of K fields of item attributes $\mathcal{B} = \{\mathcal{B}^1, \mathcal{B}^2, \ldots, \mathcal{B}^K\}$, and a set of R fields of contextual features denoted as $C = \{C^1, C^2, \ldots, C^R\}$, where item attributes include text and images information, etc. Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_I, y_I)\}$ denote a set of I user-item feedback and their corresponding labels, an instance of which can be represented as follows,

$$\mathbf{x}_i = \left[u^i, v^i, \mathbf{A}_{u^i}, \mathbf{B}_{v^i}, \mathbf{C}^i\right], \tag{1}$$

where $u^i \in \mathcal{U}$, $v^i \in \mathcal{V}$ and $\mathbf{C}^i \subset C$ denote the user, item, and context involved in the ith instance, and $\mathbf{A}_{u^i} \subset \mathcal{A}$ and $\mathbf{B}_{v^i} \subset \mathcal{B}$ are a list of attributes associated with u^i and v^i, respectively. Since only biased feedback is available, we have $\mathcal{S} \in \mathcal{P}(x, y)$, where $\mathcal{P}$ is a biased feedback distribution. It can be decomposed into two parts containing the beneficial or harmful biases, respectively, i.e., $\mathcal{P} = (\mathcal{P}^b, \mathcal{P}^h)$. Let $Q(x, y)$ denote the unbiased feedback distribution, and the goal of accuracy-bias tradeoff recommendation is to train a recommendation model based on a set of biased feedback $\mathcal{S}$, which can be adapted to a joint distribution incorporating the beneficial biases in the unbiased nature $Q^* = (\mathcal{P}^b, Q)$, so as to achieve a more ideal user experience.

4 ACCURACY-BIAS TRADEOFF RECOMMENDATION VIA CONFOUNDING PROXY PRIORS

4.1 Architecture

The confounding proxy priors-based accuracy-bias tradeoff recommendation framework, or CP2Rec for short, is shown in Figure 2. Given a current instance $(\mathbf{x}_i, y_i)$, the feature partitioning module divides the input features into two sets based on expert knowledge, including a set of potentially confounding proxy features $\mathbf{x}_i^c$ and the remaining features $\mathbf{x}_i^z$, i.e., $\mathbf{x}_i = \{\mathbf{x}_i^c, \mathbf{x}_i^z\}$, After obtaining the two feature sets, they pass through the embedding layer to obtain their respective low-dimensional dense representations, i.e., $\mathbf{c}_i$ and $\mathbf{z}_i$. Then, in the bias disentangling module, based on the guidance of confounding proxy embeddings $\mathbf{c}_i$, we can introduce the purification operator with some orthogonal constraints to encourage the model to avoid encoding the biased information in the feature embeddings $\mathbf{z}_i$ as much as possible. To be compatible with various orthogonal constraints, we use $\mathbf{z}_i'$ to denote the purified feature embeddings. In the accuracy-bias tradeoff module, we introduce an extraction operator for the confounding proxy embeddings $\mathbf{c}_i$ in an attempt to distinguish the embeddings $\mathbf{c}_i'$ driven by the beneficial bias information. To ensure that the model reasonably achieves the above goals, we additionally introduce an unbiased loss to cooperate with the original biased loss to form an effective joint training paradigm, where the unbiased loss and biased loss are responsible for the combination of the purified feature embedding with the confounding proxy embedding and the beneficial bias information embedding, respectively, i.e., $[\mathbf{c}_i, \mathbf{z}_i']$ and $[\mathbf{c}_i', \mathbf{z}_i']$. Therefore, the final optimization objective function of our CP2Rec can be expressed as follows,

$$\min_{\theta} \mathcal{L}_{CP2Rec} = \gamma \mathcal{L}_B + \mathcal{L}_U + \beta \mathcal{L}_O + \lambda \|\theta\|, \tag{2}$$

where $\mathcal{L}_B$, $\mathcal{L}_U$, and $\mathcal{L}_O$ denote the original biased loss, the unbiased loss for the accuracy-bias tradeoff module and the orthogonal constraint for the bias disentangling module, respectively, and λ and $\|\theta\|$ are the tradeoff parameter and the regularization terms. Next, we describe each module in detail based on the training process.

4.2 The Feature Partitioning Module

As shown at the bottom of Figure 2, after receiving all the input features $\mathbf{x}_i$, we need to filter out a set of potential confounding proxy features $\mathbf{x}_i^c$ based on expert knowledge. Since we focus on mitigating the system-induced biases and expect the model to preserve the beneficial bias associated with the biasedness of user decision-making behavior, we can define the selection mechanism for the confounding proxy features as follows:

- Targeted examination of statistical features associated with the items. Intuitively, the statistical features related to an item are more likely to be captured by the recommendation model, thus affecting the exposure probability of this item in the subsequent recommendation policy.
- Check the contextual feature associated with user decision-making behavior and filter out a relatively reasonable subset from it. This means that a reasonably biased decision-making behavior is also expected by the user experience, and the features related to it are more likely to have beneficial biased information and should be used as prior knowledge.

Based on the above principles, an example of the confounding proxy features used in the experiments is shown in Table 1. Note that although it is difficult to define all the features of the confounding proxy in practice, using only a subset of them can also calibrate the learning of the model well.

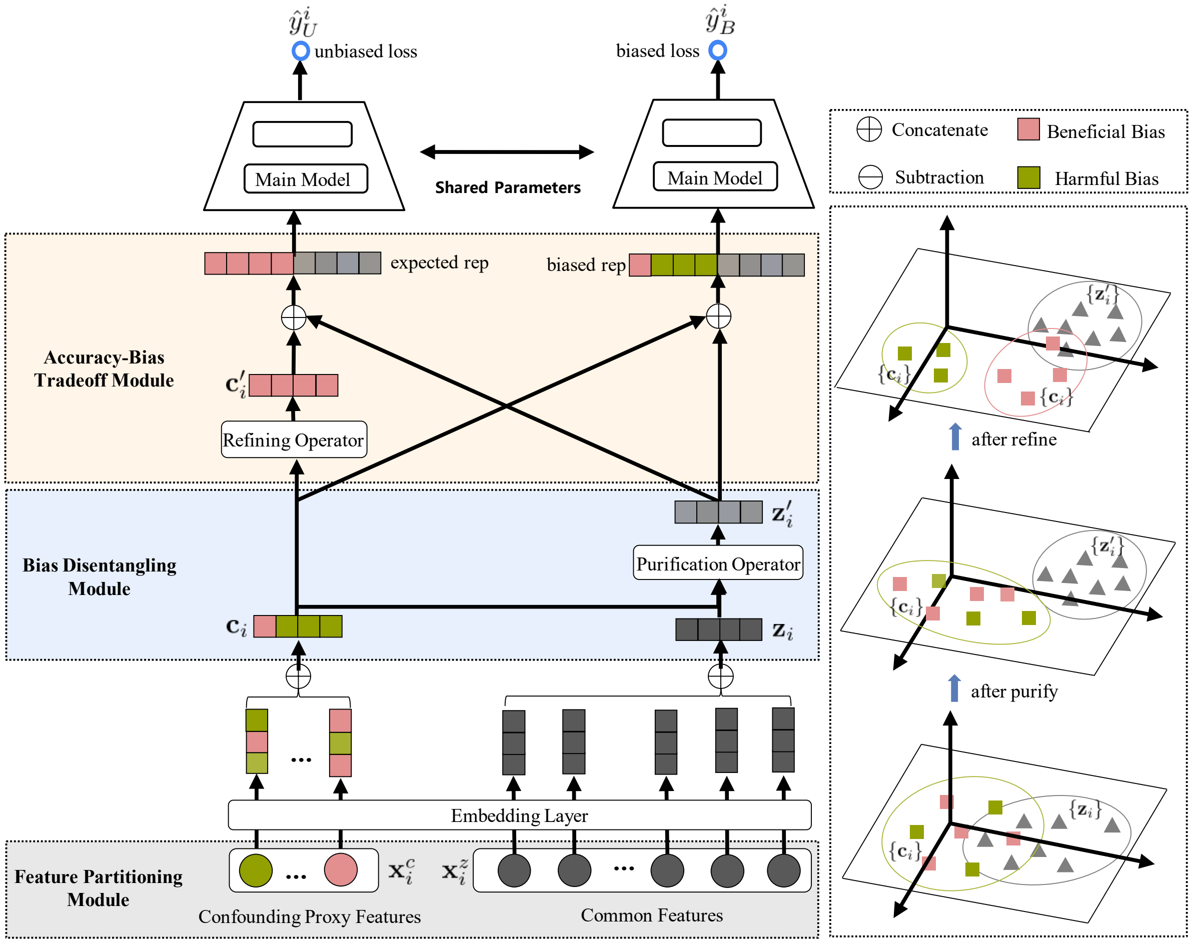


Figure 2: The architecture of the confounding proxy priors-based accuracy-bias tradeoff recommendation (CP2Rec) framework consists of three modules: 1) the feature partitioning module is used to filter out a subset of features from the input features based on expert knowledge, and this subset of features acts as the confounding proxy, explicitly responsible for the system-induced biases; 2) the bias disentangling module aims to utilize the embedding obtained by the confounding proxy to remove as much the bias-related information encoding as possible from the embedding driven by the remaining features to achieve the goal of debiasing; and 3) the accuracy-bias tradeoff module introduces an auxiliary unbiased loss that effectively synergizes with the original biased loss to induce the beneficial bias information from the embeddings of the confounding proxy and combine them with the purified feature embeddings for the inference stage.

Table 1: An example of confounding proxy features included in the online financial recommendation platform used in the experiments.

Feature Category	Example
Item-related	*item ctr, item cvr, item display style, etc.*
Context-related	*exposure position, device, market condition, etc.*

4.3 The Bias Disentangling Module

After obtaining the two feature subsets from the feature partitioning module, i.e., $\mathbf{x}_i^c$ and $\mathbf{x}_i^z$, we can use a feature embedding layer common in CTR tasks to encode their representations, i.e., $\mathbf{c}_i$ and $\mathbf{z}_i$. In this module, we propose to use a purification operator with the orthogonal constraint to explore the guidance of confounding proxy embeddings $\mathbf{c}_i$ to remove the encoding of biased information from feature embeddings $\mathbf{z}_i$. As shown on the right side of Figure 2, the idea behind this module is to obtain purified feature embeddings $\mathbf{z}_i'$ that avoid the influence of biased information. To examine the compatibility of our CP2Rec in the experiments, we consider three different implementations of orthogonal constraints in the purification operation and referred to the corresponding three variants as CP2Rec-COS, CP2Rec-PROJ, and CP2Rec-SOLVE. For ease of understanding, we give schematic diagrams of different purification operators in Figure 3.

4.3.1 Regularizer-Based Orthogonal Constraints. First, as shown on the left side of Figure 3, we consider a regularizer-based implementation. Specifically, to remove the encoding of bias information from

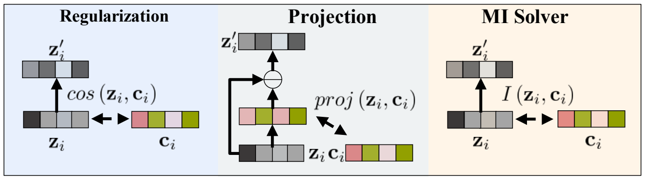


Figure 3: Schematic diagram of different purification operators.

$\mathbf{z}_i$, an intuitive idea is to constrain the orthogonality between $\mathbf{z}_i$ and $\mathbf{c}_i$, which can be equivalent to constraining the cosine similarity between two representations to approach zero,

$$cos\,(\mathbf{z}_i, \mathbf{c}_i) \rightarrow 0. \tag{3}$$

Therefore, we can get the orthogonality constraint corresponding to the purification operation, where $abs(\cdot)$ is an absolute value operation.

$$\mathcal{L}_O = \frac{1}{I}\sum_{i=1}^{I} abs\left(\frac{\mathbf{z}_i \cdot \mathbf{c}_i}{|\mathbf{z}_i|\,|\mathbf{c}_i|}\right). \tag{4}$$

4.3.2 Projection-Based Orthogonal Constraints. Second, as shown in the middle of Figure 3, we consider projection-based implementation [25]. Since directly constraining the similarity between the representations in a high-dimensional space can be difficult, projection-based operation first projects the feature embedding $\mathbf{z}_i$ into the confounding proxy embedding $\mathbf{c}_i$, so that the biased information in $\mathbf{z}_i$ can be induced,

$$\mathbf{z}_\mathbf{c} = proj\,(\mathbf{z}_i, \mathbf{c}_i) = \frac{\mathbf{z}_i \cdot \mathbf{c}_i}{|\mathbf{c}_i|} \cdot \frac{\mathbf{c}_i}{|\mathbf{c}_i|}. \tag{5}$$

We then remove this bias component $\mathbf{z}_\mathbf{c}$ from $\mathbf{z}_i$ to obtain a purified feature embedding,

$$\mathbf{z}_i' = \mathbf{z}_i - \mathbf{z}_\mathbf{c}. \tag{6}$$

Note that in this case there is no explicit orthogonality constraint loss $\mathcal{L}_O$, and the purified feature embeddings $\mathbf{z}_i'$ are updated through subsequent loss functions.

4.3.3 Orthogonal Constraints with Mutual Information Solver. Finally, as shown on the right side of Figure 3, we consider implementation based on mutual information solver. Based on the perspective of information theory, the uncorrelation between two representations means that the mutual information between them is zero [31].

$$I\,(\mathbf{z}_i, \mathbf{c}_i) \rightarrow 0. \tag{7}$$

Therefore, we can directly use some existing mutual information solvers to implement this constraint, and we use CLUB [6] as an example in our experiments, where $q_\theta(\cdot|\cdot)$ is a variational distribution.

$$\mathcal{L}_O = \frac{1}{I}\sum_{i=1}^{I}\left[\log q_\theta(\mathbf{c}_i|\mathbf{z}_i) - \frac{1}{I}\sum_{j=1}^{I}\log q_\theta(\mathbf{c}_j|\mathbf{z}_i)\right]. \tag{8}$$

4.4 The Accuracy-Bias Tradeoff Module

After the bias disentangling module, the feature embedding $\mathbf{z}_i$ is purified to avoid encoding the biased information as much as possible. For the convenience of description, we denote the feature embedding at this time as $\mathbf{z}_i'$, and note that since the feature embedding is not explicitly modified in Section 4.3.1 and 4.3.3, there is $\mathbf{z}_i' = \mathbf{z}_i$. Since only biased feedback is available during training, an original loss function $\mathcal{L}_B$ is biased, and we use a combination of the confounding proxy embeddings and the purified feature embeddings for prediction,

$$\hat{y}_B^i = f\left([\mathbf{c}_i, \mathbf{z}_i'], \theta\right), \tag{9}$$

$$\mathcal{L}_B = \mathcal{L}(\hat{y}_B^i, y_i), \tag{10}$$

where f is a predictive mapping function and θ is the model parameters. To recover the beneficial bias information in the feature embeddings, in this module, we propose a refining module to capture the beneficial bias information driven by the confounding proxy embeddings,

$$\mathbf{c}_i' = g\,(\mathbf{c}_i) = \sigma(\mathbf{W} * \mathbf{c}_i + \mathbf{b}), \tag{11}$$

where $\mathbf{W}$ is a weight matrix, $\mathbf{b}$ is a bias vector, and $\sigma(\cdot)$ is an activation function. To constrain the $\mathbf{c}_i'$ obtained by Equation 11 to accurately capture the beneficial bias information, we introduce an auxiliary unbiased loss function $\mathcal{L}_U$ and make predictions by combining $\mathbf{c}_i'$ with purified feature embeddings $\mathbf{z}_i'$,

$$\hat{y}_U^i = f\left([\mathbf{c}_i', \mathbf{z}_i'], \theta\right). \tag{12}$$

$$\mathcal{L}_U = \mathcal{L}(\hat{y}_U^i, y_i). \tag{13}$$

Note that many unbiased loss functions have been proposed in existing work, and we use unbiased optimization with inverse propensity score (IPS) [28] as an example in our experiments. We synergize the two loss functions by sharing model parameters to accurately identify the desired $\mathbf{c}_i'$ from the confounding proxy embeddings $\mathbf{c}_i$. The idea behind this module is to not only guarantee the correct learning of purified feature embeddings $\mathbf{z}_i'$ and confounding proxy embeddings $\mathbf{c}_i$ but also extract information from the latter that is beneficial to predict users' biased decisions in unbiased scenarios. Furthermore, we use Equation 12 as the final prediction at the inference stage to achieve the accuracy-bias tradeoff.

5 EMPIRICAL EVALUATIONS

In this section, we conduct experiments with the aim of answering the following four key questions.

- RQ1: How does our CP2Rec perform compared to the baselines?
- RQ2: What is the role of each module in our CP2Rec?
- RQ3: What are the characteristics of the representations obtained in our CP2Rec?
- RQ4: How effective is our CP2Rec in an online deployment?

5.1 Experimental Setup

5.1.1 Datasets. In order to evaluate the ideal performance of the model in alleviating the system-induced biases, we need a test set containing unbiased feedback to simulate the ideal feedback distribution. To our best knowledge, there is only one publicly available dataset in the existing literature that satisfies this property with the required multimodal features, i.e., KuaiRec[1] [8]. KuaiRec is collected from the recommendation logs of Kuaishou[2], a Chinese video-sharing platform, which contains a traditionally biased feedback subset and an almost fully-observed feedback subset that can

[1]https://kuairec.com/
[2]https://www.kuaishou.com/en

approximate an unbiased distribution. For a more comprehensive evaluation, we also use a real product dataset in our experiments, i.e., Product. The Product is a subset sampled from the log data collected from the homepage recommendation business of a large-scale online financial management platform in China. Depending on the collection process, it also contains a biased feedback subset and an unbiased feedback subset and provides multimodal features such as text and images.

5.1.2 Dataset Preprocessing. Following the settings of previous works [17, 18], we split the biased feedback subset into training and validation sets in a ratio of 8:2 to strictly obey the case where an unbiased feedback subset is not available. We use the entire unbiased feedback subset as the test set for unbiased evaluation of the model. In addition, we select 11 and 14 features from the item-related statistical feature set as the confounding proxy features, respectively. We summarize the statistics of the two processed datasets in Table 2.

Table 2: Statistics of the processed datasets.

Dataset	KuaiRec	Product
#User	7,176	1,020,345
#Item	10,728	1,549
#Training Instance	10,024,644	8,820,589
#Validation Instance	2,506,162	2,192,537
#Test Instance	4,676,570	1,168,594
#User Attribute	31	349
#Item Attribute	57	176
#Contextual Feature	-	29
#Confounding Proxy Feature	11	14

5.1.3 Baselines. We select the representative methods among the three routes of debiased learning without an unbiased feedback subset, which are summarized in Section 2. For the first route, we use the inverse propensity score (IPS) [29] and the self-normalized IPS (SNIPS) [30] as our baselines since they are the most common baseline candidates in debiasing learning. For the second route, we select the relevance-maximization optimization (Rel) [28] as our baseline due to its demonstrated performance benefits in previous works [17, 26]. For the third route, we choose two recent representative methods as our baselines, i.e., CVIB [36] and DIB [18]. CVIB uses the information bottleneck to constrain the model to learn the balanced feature information between the observed and unobserved feedback, and DIB prompts the model to accurately distinguish the information required by the biased and unbiased components from the input features, respectively. Since most debiasing methods are model-agnostic, in order to evaluate the generalization of all the methods, we adopt two of the most common CTR models as the backbones, i.e., neural collaborative filtering (NCF) [10] and deep & cross network (DCN) [10], i.e., each of these methods has two corresponding versions based on different backbones.

5.1.4 Evaluation Metrics. Following the setup employed in previous works [14, 39], we evaluate the performance of CTR prediction via two widely used metrics, i.e., the area under the ROC curve (AUC) and the negative logarithmic loss (NLL). We choose AUC as our main evaluation metric because it is one of the most important metrics in the industry and previous works on debiasing. Note that for both metrics, higher values indicate better results.

5.1.5 Implementation Details. We implement all the methods on TensorFlow 1.15[3] with the Adam optimizer and the embedding dimension fixed at 10. In order to speed up the tuning process, we use the Bayesian optimization library *Optuna*[4] [1] for the dynamic search of hyperparameters. We also adopt an early stopping mechanism with patience of 5 to avoid overfitting to the training set. The range of the values of the hyper-parameters is shown in Table 3.

Table 3: Hyper-parameters tuned in the experiments.

Name	Range	Functionality
λ	$\{1e^{-6}, 1e^{-5} \cdots 1e^{-2}\}$	regularization
lr	$\{1e^{-4}, 5e^{-4}, 1e^{-3} \cdots 1e^{-2}\}$	learning rate
bs	$\{2^7, 2^8, \cdots 2^{10}\}$	batch size
γ	$\{1e^{-5}, 1e^{-5} \cdots 1e^{-1}\}$	loss weight
β	$\{1e^{-5}, 1e^{-5} \cdots 1e^{-1}\}$	loss weight

5.2 RQ1: Performance Comparison

We report the overall performance comparison in Table 4 and 5. From the results in Table 4 and 5, we can have the following observations: 1) All the debiasing methods perform better than the original backbone. For example, DIB-NCF increase AUC by 0.02 compared with NCF on KuaiRec. This further necessitates our research here; and 2) Compared with other baselines, our CP2Rec performs best despite different datasets and backbones in both AUC and NLL. With different purification operators, CP2Rec performs consistently well compared with other debiasing baselines. However, the best operator varies with the dataset and backbone. Specifically, with NCF, CP2Rec-PROJ is superior to other methods on the KuaiRec dataset, while CP2Rec-SOLVE is the best on the product dataset. With DCN, CP2Rec-COS is the best on KuaiRec dataset, and CP2Rec-PROJ achieves better on the product dataset. One possible reason for this discrepancy is that when the backbone model is relatively simple, no matter what degree of purification operator can be trained well, thus more complex variants (e.g., PROJ or SOLVE) can show a better result. But when the backbone model is more complex, some complex purification operators may sometimes not be well trained, and relatively simple purification operators may perform better. Note that all the purification operators are easily plug-and-play in our framework, which verifies the effectiveness of our framework.

5.3 RQ2: Ablation Study of CP2Rec

In this subsection, we conduct the ablation study over the modules of our framework, which is demonstrated in Table 6. Here, we compare it with other two methods: (i) w/o ABT, which is the CP2Rec-COS-NCF model without accuracy-bias tradeoff module; (ii) w/o ABT, BD, which is the model with neither accuracy-bias tradeoff

[3] https://www.tensorflow.org
[4] https://optuna.org/

Table 4: Comparison results of unbiased evaluation using NCF as the backbone model, where the best results and the second best results are marked in bold and underlined, respectively. Note that AUC is the main evaluation metric, and * indicates a significance level of $p \leq 0.05$ based on a two-sample t-test between our method and the best baseline.

Dataset	KuaiRec		Product	
Metrics	**AUC↑**	**NLL↑**	**AUC↑**	**NLL↑**
NCF	0.8246	-0.6899	0.8601	-0.6532
IPS-NCF	0.8273	-0.6890	0.8685	-0.6538
SNIPS-NCF	0.8144	-0.6898	0.8642	-0.6522
CVIB-NCF	0.8265	-0.6897	0.8637	-0.6539
Rel-NCF	0.8388	-0.6860	0.8804	-0.6502
DIB-NCF	0.8456	-0.6902	0.8835	-0.6503
CP2Rec-COS-NCF	0.8483	-0.6861	0.8860	-0.6415
CP2Rec-PROJ-NCF	**0.8492***	**-0.6843***	<u>0.8876</u>	<u>-0.6381</u>
CP2Rec-SOLVE-NCF	<u>0.8491</u>	<u>-0.6843</u>	**0.8898***	**-0.6375***

Table 5: Comparison results of unbiased evaluation using DCN as the backbone model, where the best results and the second best results are marked in bold and underlined, respectively. Note that AUC is the main evaluation metric, and * indicates a significance level of $p \leq 0.05$ based on a two-sample t-test between our method and the best baseline.

Dataset	KuaiRec		Product	
Metrics	**AUC↑**	**NLL↑**	**AUC↑**	**NLL↑**
DCN	0.8257	-0.6900	0.8711	-0.6538
IPS-DCN	0.8239	-0.6903	0.8744	-0.6524
SNIPS-DCN	0.8028	-0.6905	0.8759	-0.6517
CVIB-DCN	0.8241	-0.6898	0.8776	-0.6507
Rel-DCN	0.8311	-0.6885	0.8838	-0.6495
DIB-DCN	0.8455	-0.6891	0.8854	-0.6480
CP2Rec-COS-DCN	**0.8478***	<u>-0.6849</u>	<u>0.8918</u>	<u>-0.6362</u>
CP2Rec-PROJ-DCN	0.8461	-0.6858	**0.8922***	-0.6445
CP2Rec-SOLVE-DCN	<u>0.8474</u>	**-0.6846***	0.8903	**-0.6352***

module nor bias disentangling module. From the results, we can observe that without ABT, the AUC decreased by 0.0022, while AUC decreased by 0.0028 without ABT and BD. The observation validates the effectiveness of the designed modules in the framework.

5.4 RQ3: Visualization of the Representations

To further verify whether our CP2Rec can effectively distinguish between biased and unbiased information, we use t-SNE [32] to visualize the obtained representations and observe their characteristics. We demonstrate the representations for all the instances on the Product dataset using NCF as the backbone setting. As shown in Figure 4, no matter which variant, there is a clear boundary between the blue points representing the unbiased components z and the red points representing the biased components c. This indicates that our CP2Rec can accurately capture both the biased components and unbiased components. Also, note that some of the red points are closer to the blue region than the rest of the red, which may be the beneficial bias identified by the model that is expected to be preserved.

Table 6: Results of the ablation studies on KuaiRec using NCF as the backbone setting, where the best results are marked in bold.

Dataset	KuaiRec		
Method	CP2Rec-COS-NCF	w/o ABT	w/o ABT, BD
AUC↑	**0.8483**	0.8461	0.8456
NLL↑	**-0.6861**	-0.6870	-0.6902

5.5 RQ4: Results of the Online Deployment

We deploy our CP2Rec in an online financial recommendation platform, which serves millions of daily active users. All the models are trained in a single cluster, where each node contains 96 core Intel(R) Platinum 8255C CPU, 256GB RAM as well as 8 NVIDIA TESLA A100 GPU cards. We conduct the A/B test for three weeks, and two commonly used online evaluation metrics, CLPM (i.e., $\frac{\#click}{\#impression}$), COPM (i.e., $\frac{\#conversion}{\#impression}$), are used. Besides, we also use purchase amount per mile (PAPM) defined as $\frac{\#amount}{\#impression}$ to evaluate the model in the financial recommendation platform. The daily improvements are illustrated in Figure 5. On average, we get improvements in all three metrics as shown in Table 7, which further verifies the effectiveness of our method. Moreover, we also investigate how the exposure probability of the top 40 popular items varies online. From Figure 6, we observe that items in the top list are supposed to be suppressed popularity, while others are given more impressions leading to alleviating the bias problem in this industry recommender system.

Table 7: Average improvement of our CP2Rec in a three-week online A/B test.

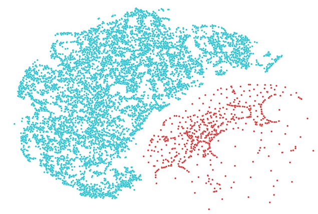
Method	CTPM (imp.)	COPM (imp.)	PAPM (imp.)
CP2Rec-COS	2.47%	5.10%	5.74%
CP2Rec-PROJ	2.89%	6.19%	5.88%
CP2Rec-SOLVE	2.91%	7.03%	7.97%

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we introduce the confounding proxy features as prior knowledge for model training in the debiased recommendation and propose a novel debiasing framework, CP2Rec. In our CP2Rec, we first leverage these confounding proxy features in the bias disentangling module to guide the model to learn the bias-independent

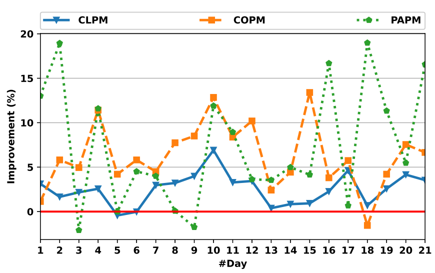
(a) CP2Rec-COS

(b) CP2Rec-PROJ

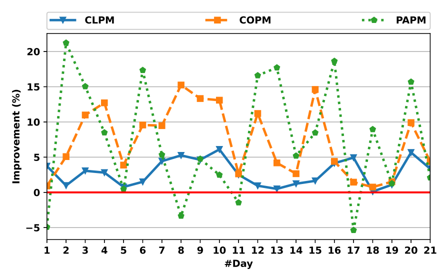


(c) CP2Rec-SOLVE

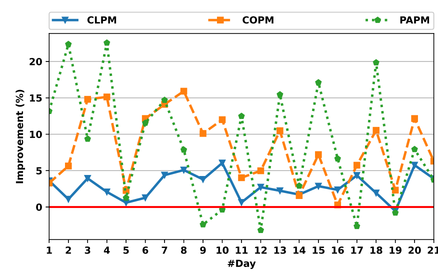
Figure 4: Visualization of the biased component c and the unbiased component z, in which the red and blue points are used to denote the biased component c and the unbiased component z, respectively.



(a) CP2Rec-COS

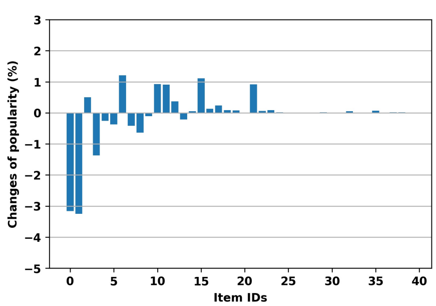


(b) CP2Rec-PROJ

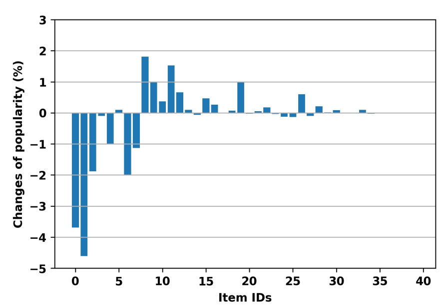


(c) CP2Rec-SOLVE

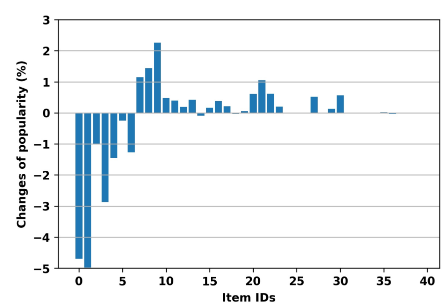
Figure 5: Improvements of our CP2Rec variants compared with the base model in the online A/B test, including total clicks per mille (CLPM), total conversions per mille (COPM) and purchase amount per mille (PAPM).



(a) CP2Rec-COS



(b) CP2Rec-PROJ



(c) CP2Rec-SOLVE

Figure 6: Difference in the exposure probability of the top 40 popular items after deploying our CP2Rec, where the item IDs on the x-axis have been sorted in descending order according to their popularity.

feature embeddings. Then, in the accuracy-bias tradeoff module, we recover the beneficial bias information from the feature embeddings with the help of an unbiased loss, thereby achieving a more reasonable accuracy-bias tradeoff for the multimedia recommendation. Finally, we conduct extensive experiments on a public dataset, a real product dataset, and an online AB test to demonstrate the effectiveness of our CP2Rec.

For future work, we plan to design some automated selection mechanisms for the confounding proxy features. We also plan to explore more orthogonal constraints between the representations to further improve the effectiveness of removing bias effects. In addition, we are also interested in trying some new strategies to assist the model in recovering beneficial bias information.

ACKNOWLEDGMENTS

We thank the support of the National Natural Science Foundation of China Nos. 61836005, 62272315, and 62172283.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2623–2631.

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

[3] Feiyu Chen, Junjie Wang, Yinwei Wei, Hai-Tao Zheng, and Jie Shao. 2022. Breaking isolation: Multimodal graph fusion for multimedia recommendation by edge-wise modulation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 385–394.

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

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[6] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. CLUB: A contrastive log-ratio upper bound of mutual information. In *Proceedings of the 37th International Conference on Machine Learning*. 1779–1788.

[7] 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*. 40–49.

[8] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 540–550.

[9] Prem Gopalan, Jake M Hofman, and David M Blei. 2015. Scalable recommendation with hierarchical Poisson factorization. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*. 326–335.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the ACM Web Conference 2017*. 173–182.

[11] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, and Peng Wu. 2023. Balancing unobserved confounding with a few unbiased ratings in debiased recommendations. In *Proceedings of the ACM Web Conference 2023*. 1305–1313.

[12] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the ACM Web Conference 2016*. 951–961.

[13] Chen Lin, Dugang Liu, Hanghang Tong, and Yanghua Xiao. 2022. Spiral of silence and its application in recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 34, 6 (2022), 2934–2947.

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

[15] Dugang Liu, Pengxiang Cheng, Zinan Lin, Jinwei Luo, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2022. KDCRec: Knowledge distillation for counterfactual recommendation via uniform data. *IEEE Transactions on Knowledge and Data Engineering* (2022). https://doi.org/10.1109/TKDE.2022.3199585

[16] Dugang Liu, Pengxiang Cheng, Zinan Lin, Xiaolian Zhang, Zhenhua Dong, Rui Zhang, Xiuqiang He, Weike Pan, and Zhong Ming. 2023. Bounding system-induced biases in recommender systems with a randomized dataset. *ACM Transactions on Information Systems* 41, 4 (2023), 1–26.

[17] 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 *Proceedings of the 15th ACM Conference on Recommender Systems*. 351–360.

[18] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2023. Debiased representation learning in recommendation via information bottleneck. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–27.

[19] Dugang Liu, Chen Lin, Zhilin Zhang, Yanghua Xiao, and Hanghang Tong. 2019. Spiral of silence in recommender systems. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 222–230.

[20] Dugang Liu, Yang Qiao, Xing Tang, Liang Chen, Xiuqiang He, Weike Pan, and Zhong Ming. 2023. Self-sampling training and evaluation for the accuracy-bias tradeoff in recommendation. In *International Conference on Database Systems for Advanced Applications*. 580–592.

[21] David C. Liu, Stephanie Rogers, Raymond Shiau, Dmitry Kislyuk, Kevin C. Ma, Zhigang Zhong, Jenny Liu, and Yushi Jing. 2017. Related pins at pinterest: The evolution of a real-world recommender system. In *Companion Proceedings of the ACM Web Conference 2017*. 583–592.

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

[23] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. 5–12.

[24] Zongshen Mu, Yueting Zhuang, Jie Tan, Jun Xiao, and Siliang Tang. 2022. Learning hybrid behavior patterns for multimedia recommendation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 376–384.

[25] Qi Qin, Wenpeng Hu, and Bing Liu. 2020. Feature projection for improved text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8161–8171.

[26] Yi Ren, Hongyan Tang, Jiangpeng Rong, and Siwen Zhu. 2023. Unbiased pairwise learning from implicit feedback for recommender systems without biased variance control. *arXiv preprint arXiv:2304.05066* (2023).

[27] Yuta Saito. 2020. Asymmetric tri-training for debiasing missing-not-at-random explicit feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 309–318.

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

[29] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on Machine Learning*. 1670–1679.

[30] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *Proceedings of the 29th International Conference on Neural Information Processing Systems*. 3231–3239.

[31] Naftali Tishby, Fernando C Pereira, and William Bialek. 1999. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communications, Control and Computing*. 368–377.

[32] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008).

[33] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1288–1297.

[34] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *Proceedings of the 36th International Conference on Machine Learning*. 6638–6647.

[35] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2021. Combating selection biases in recommender systems with a few unbiased ratings. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 427–435.

[36] Zifeng Wang, Xi Chen, Rui Wen, Shao-Lun Huang, Ercan E Kuruoglu, and Yefeng Zheng. 2020. Information theoretic counterfactual learning from missing-not-at-random feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 1854–1864.

[37] 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 *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1791–1800.

[38] Haiqin Yang, Guang Ling, Yuxin Su, Michael R Lyu, and Irwin King. 2015. Boosting response aware model-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering* 27, 8 (2015), 2064–2077.

[39] Bowen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. 2019. Improving ad click prediction by considering non-displayed events. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 329–338.

[40] Shuxi Zeng, Murat Ali Bayir, Joseph J Pfeiffer III, Denis Charles, and Emre Kiciman. 2021. Causal transfer random forest: Combining logged data and randomized experiments for robust prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 211–219.

[41] Xiaoying Zhang, Junzhou Zhao, and John CS Lui. 2017. Modeling the assimilation-contrast effects in online product rating systems: Debiasing and recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 98–106.

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