

Causal Distillation for Alleviating Performance Heterogeneity in Recommender Systems

Shengyu Zhang[†], Ziqi Jiang[†], Jiangchao Yao, Fuli Feng, Kun Kuang*, Zhou Zhao, Shuo Li, Hongxia Yang, Tat-seng Chua, Fei Wu*, Senior Member, IEEE

Abstract—Recommendation performance usually exhibits a long-tail distribution over users — a small portion of head users enjoy much more accurate recommendation services than the others. We reveal two sources of this performance heterogeneity problem: the uneven distribution of historical interactions (a natural source); and the biased training of recommender models (a model source). As addressing this problem cannot sacrifice the overall performance, a wise choice is to eliminate the model bias while maintaining the natural heterogeneity. The key to debiased training lies in eliminating the effect of confounders that influence both the user's historical behaviors and the next behavior. The emerging causal recommendation methods achieve this by modeling the causal effect between user behaviors, however potentially neglect unobserved confounders (e.g., friend suggestions) that are hard to measure in practice. To address unobserved confounders, we resort to the front-door adjustment (FDA) in causal theory and propose a causal multi-teacher distillation framework (CausalD). FDA requires proper mediators in order to estimate the causal effects of historical behaviors on the next behavior. To achieve this, we equip CausalD with multiple heterogeneous recommendation models to model the mediator distribution. Then, the causal effect estimated by FDA is the expectation of recommendation prediction over the mediator distribution and the prior distribution of historical behaviors, which is technically achieved by multi-teacher ensemble. To pursue efficient inference, CausalD further distills multiple teachers into one student model to directly infer the causal effect for making recommendations. We instantiate CausalD on two representative models, DeepFM and DIN, and conduct extensive experiments on three real-world datasets, which validate the superiority of CausalD over state-of-the-art methods. Through in-depth analysis, we find that CausalD largely improves the performance of tail users, reduces the performance heterogeneity, and enhances the overall performance.

Index Terms—Recommender System; Performance Heterogeneity; Causal Distillation; Front-door Adjustment

1 INTRODUCTION

Recommender systems alleviate the information overload on the Web by providing personalized services for users seeking information. It has become a default choice to learn recommender models from user historical behaviors [2], [3], [4], [5]. The model typically exhibits performance heterogeneity with significant divergence across different user groups. Figure 1 provides empirical evidence where we train a representative recommendation model DIN [1] on the benchmark Amazon Review dataset [6] and evaluate the performance over equally sized user groups. User groups

- Shengyu Zhang, Ziqi Jiang, and Zhou Zhao are with Zhejiang University, Hangzhou 310027, China. E-mail: {sy_zhang, jiangzq, zhouchao}@zju.edu.cn
- Kun Kuang is with Zhejiang University, Hangzhou 310027, China, and also with the Key Laboratory for Corneal Diseases Research of Zhejiang Province, Hangzhou 310016, China. E-mail: {kunkuang}@zju.edu.cn
- Fei Wu is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, with the Shanghai Institute for Advanced Study, Zhejiang University, Shanghai 201210, China, and also with the Shanghai AI Laboratory, Shanghai 200232, China. E-mail: wufei@zju.edu.cn
- Jiangchao Yao is with the Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: Sunarker@sjtu.edu.cn
- Fuli Feng is with the University of Science and Technology of China, Hefei 230026, China. E-mail: fulifeng93@gmail.com
- Shuo Li and Tat-Seng Chua are with the National University of Singapore, Singapore. E-mail: li.shuo980@gmail.com, chuats@comp.nus.edu.sg
- Hongxia Yang is with Alibaba Group, Hangzhou 310052, China. E-mail: yang.yhx@alibaba-inc.com

[†]Equal contribution

*Corresponding authors

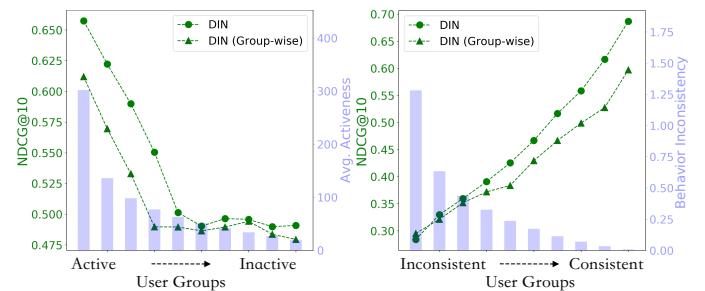


Fig. 1: Recommendation performance over user groups clustered by 1) user activeness, and 2) behavior consistency on popular items. DIN [1] and DIN (Group-wise) are under unified training and group-wise training, respectively. The user number of each group remains the same.

can be divided according to diverse factors, and we take two factors for illustration: 1) the activeness of users (*i.e.*, the number of interactions during training); and 2) the user behavior consistency on popular items¹. In both cases, it is observed that a small portion of head users receive more accurate recommendation results than the others. Moreover, head users enjoy significantly more performance

¹We cluster items into eighteen groups by popularity, equally split the behavior sequence of each user into two parts (past and now), and represent each part with item distribution over the clusters. We measure the behavior consistency on popular items of a user with the KL divergence of these two distributions. Intuitively, recommender models will perform better on users with consistent behaviors.

gains under unified training (one model for all groups) than group-wise training (one model for each group). This means that the unified training might aggravate the **performance heterogeneity** problem among users due to model biases. In this work, we analyze and address the performance heterogeneity issue in recommender systems, which has thus far received little scrutiny.

There are indeed two sources of the performance heterogeneity issue from the data and model perspectives. A *natural source* is the imbalance of training data distribution over users. For instance, active users are likely to receive relatively accurate recommendations owing to rich interaction records and more comprehensive interest modeling. A *model source* is that recommender models further amplify the impact of data imbalance during training [7], *i.e.*, the model bias towards the head users. Accordingly, we set two targets in addressing this performance heterogeneity issue: (1) we want to preserve the natural heterogeneity since forcibly removing it might hurt the overall performance [8]; and (2) we would like to eliminate the model bias that amplifies the heterogeneity. However, most existing methods either neglect or eliminate the heterogeneity, such as the fairness recommendation [9] that pursues equal performance across different groups².

To achieve the targets, we dive into the generation process of interactions to uncover the causes of the model bias, which is abstracted as a *causal graph* [10] as depicted in Figure 2:

- $X \rightarrow Y$. Recommendation models predict user next behavior Y (*e.g.*, clicking an item), based on user historical behaviors X under the assumption that historical behaviors X are the cause of next behavior Y [2].
- $U \rightarrow X, U \rightarrow Y$. U represents a set of factors (*e.g.*, user activeness, and user preference on popular items) that directly influence user historical behaviors and next behaviors apart from the effect $X \rightarrow Y$. For instance, user activeness will increase the size of historical records $U \rightarrow X$ and encourage item exploration $U \rightarrow Y$. User preference on popular items will influence the distribution of user historical behaviors $U \rightarrow X$ and the property of the next item $U \rightarrow Y$.

U is identified as a confounder between X and Y , which will lead to spurious correlation $X \leftarrow U \rightarrow Y$ when recommendation models (*e.g.*, sequential recommendation [11], [12] and graph-based recommendation [13]) estimate the correlation between historical behaviors and next behaviors [10]. The spurious correlation is the source of model bias that amplifies the performance heterogeneity. For instance, preferring popular items $U = 1$ will lead to next behaviors Y linked with popular items that are not caused by historical behaviors X . Such $<$ historical behaviors, next behavior $>$ pairs are spuriously correlated and bias the model to recommend popular items regardless of historical behaviors. Note that such model bias might not do harm to users that prefer popular items (head users). However, other users might suffer from these spurious correlations under

²Note that recommendation methods enforcing fairness also requires known sensitive features, which might be unavailable in the performance heterogeneity problem.

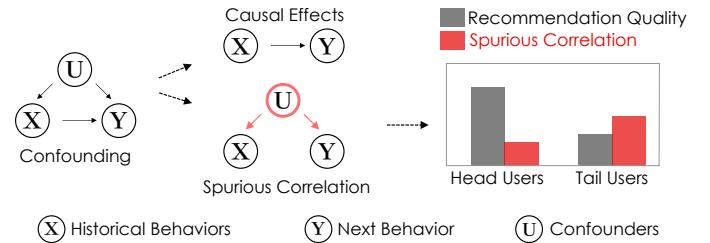


Fig. 2: Causal graph for illustrating the amplified performance heterogeneity due to spurious correlation.

the unified training³ and receive low-quality recommendations (*cf.* Figure 2). As such, the performance heterogeneity issue is amplified due to the confounding.

In this light, we focus on alleviating performance heterogeneity by blocking the spurious correlation $X \leftarrow U \rightarrow Y$, *i.e.* modeling the causal effect $X \rightarrow Y$. Intuitively, we can achieve the target by applying the debiasing techniques in recommendation that handles confounders such as inverse propensity scoring (IPS) [14], [15], back-door adjustment [16], [17], counterfactual inference [18], [19]. To obtain accurate estimation, these methods require either unbiased data or *observations of confounders*, which might not be easily satisfied in real-world recommender systems. For example, collecting unbiased data from random trials (*e.g.* uniform exposure) is costly due to hurting user experience. Meanwhile, *unobserved* confounders are common in recommendation, breaking the unconfoundedness assumption [20]. For instance, social relationships (*e.g.*, friends and family members) are known to have significant impact on recommender systems, *i.e.*, bringing exposure to items ($U \rightarrow Y$) when friends directly recommend items, and conformity to the users ($U \rightarrow X$). However, whether a click happens due to the user's inherent interests or the suggestions/recommendations from the user's friends or family members is not routinely available for many recommender systems. Another example of unobserved confounder is user's current mood, which would TODO

Hence, there is an urgent need for a debiasing method to deal with unobserved confounders without costly unbiased data.

Towards the goal, we propose to estimate the causal effect $P(Y | do(X))$ in recommendation through the front-door adjustment [10] (FDA) according to the latent representation of user historical behaviors, which is a mediator between X and Y (see Figure 3). The core idea of FDA is to estimate the outcome of the intervention $do(X)$ by adjusting the distribution $P(Y | do(M))$ over $P(M | X)$. FDA thus takes a two-step estimation procedure: 1) sampling $M = m$ from $P(M | X)$; and 2) estimating the causal effect from M to Y , *i.e.* $P(Y | do(M = m))$. However, these estimations are non-trivial in recommendation for two reasons: 1) the deterministic encoder in many existing recommendation models can only generate one latent user representation m , failing to model the conditional distribution $P(M | X)$

³The spurious correlation scores in Figure 2 are correlated with the results in Figure 1 where the experiments are conducted in the Amazon Book dataset. Tail users obtain few gains in unified training due to suffering more spurious correlations.

and sample multiple m ; and 2) under the strict latency restrictions in recommender systems, the computation cost of enumerating M and X is unaffordable during model inference. To tackle these two challenges, we encapsulate multi-teacher distillation into the instantiation of FDA, and propose the causal distillation framework, named *CausalD*. CausalD employs the pre-trained heterogeneous models to perform sampling of $M = m$ given X (Challenge 1). Upon the model-based sampling, CausalD estimates the causal effects based on user sampling and multi-teacher ensemble. The estimated causal effects serve as the distillation label for training an efficient student recommendation model, which directly estimates the effect without modeling $P(M | X)$ and $(Y | do(M = m))$ per m (Challenge 2). In summary, this work makes the following contributions:

- We analyze the performance heterogeneity issue of recommender models from a causal view and address the issue by modeling causal effect with front-door adjustment to handle the *unobserved confounders*.
- We propose a *causal multi-teacher distillation* (CausalD) framework, which realizes FDA to estimate the causal effect and preserves the inference efficiency.
- We instantiate CausalD on DIN and DeepFM, and conduct extensive experiments on three real-world datasets, validating the effectiveness of our analysis and CausalD.

2 PRELIMINARIES

Confounding is the distortion of the association between two variables X and Y , which share a common cause, namely the confounder U . The confounder will bring the spurious correlation between X and Y ($X \leftarrow U \rightarrow Y$), leading to the gap between $P(Y | X)$ and the causal effect of X on Y . To accurately estimate the causal effect, causal inference performs interventions by introducing the *do*-operator and estimates $P(Y | do(X))$ instead of $P(Y | X)$. There are mainly two deconfounding techniques that estimate $P(Y | do(X))$ [10] from observed data, *i.e.*, back-door adjustment, and front-door adjustment. In this section, we review the main idea of these techniques:

2.1 Back-door Adjustment

Back-door adjustment can handle observed confounders Z . Note that U represents all confounders. Under Bayes Rule, the posterior $P(Y | X)$ can be written as:

$$P(Y | X) = \sum_z P(Y | X, Z = z) \underline{P(Z = z | X)}. \quad (1)$$

In this modeling, there is a back-door path between X and Y , which refers to the indirect path from X to Y that contains an arrow to X , *i.e.*, $X \leftarrow Z \rightarrow Y$. To block the back-door path, back-door adjustment adjusts the parent variables of X to make these variables independent of X with the *do*-operator. As such, $X \leftarrow Z$ is cut due to the independence, *i.e.*, forcibly assigning a target value to X regardless of its parent Z . Formally,

$$\begin{aligned} P(Y | do(X)) &= \sum_z P(Y | do(X, Z = z)) P(Z = z | do(X)), \\ &= \sum_z P(Y | X, Z = z) \underline{P(Z = z)}. \end{aligned} \quad (2)$$

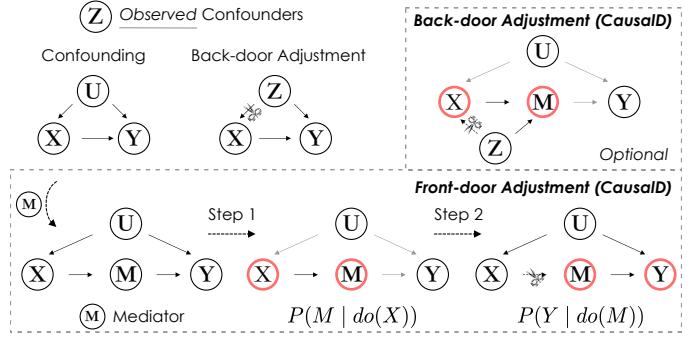


Fig. 3: Causal graph for illustrating the back-door adjustment and front-door adjustment.

The conditional probability $P(Y | X, Z = z)$ is invariant to adjustment since how Y responds to X and Z is regardless of whether the change of X is independent of Z . As such, $P(Y | do(X, Z = z)) = P(Y | X, Z = z)$. In essence, back-door adjustment computes the association between X and Y for each value z of Z , and takes the average over the prior probability $P(Z = z)$. The transition from $P(Z = z | X)$ to $P(Z = z)$ indicates that the change of X no longer affects the distribution of choosing Z , thus obeying the randomized experiments where we can accurately estimate the causal effect of X on Y .

2.2 Front-door Adjustment

From Equation (2), we can find that the back-door adjustment requires the knowledge of the confounder variable Z , *i.e.*, the detailed values z . However, there are many unobserved confounders in U that are hard to measure in practice, such as the social relationships that might bring exposure to items (*e.g.*, friends recommending items) and conformity to users, in recommender systems. Fortunately, front-door adjustment can estimate the causal effects by introducing a proper mediator variable M between X and Y . By Bayes Rule, with the mediator, the posterior with *do*-operator can be written as:

$$P(Y | do(X)) = \sum_m P(M = m | do(X)) P(Y | do(M = m)). \quad (3)$$

Then, FDA can be decomposed into the following two steps:

- 1) Estimating $P(M = m | do(X))$. $P(M = m | do(X)) = P(M = m | X)$ since the back-door path $X \leftarrow U \rightarrow Y \leftarrow M$ is naturally blocked thanks to the colliding effect $U \rightarrow Y \leftarrow M$. An uncontrolled collider variable Y will block the effect from U to M [10].
- 2) Estimating $P(Y | do(M = m))$ *w.r.t.* the unblocked back-door path $M \leftarrow X \leftarrow U \rightarrow Y$. Similar to the back-door adjustment, the remedy is to adjust the parents of M , *i.e.*, X and make (X, M) independent so that we can block the back-door path by cutting $M \leftarrow X$.

By the Bayes Rule, the original likelihood can be written as:

$$\begin{aligned} P(Y | X) &= \\ &\sum_m P(M = m | X) \sum_x \underline{P(X = x | M = m)} [P(Y | M = m, X = x)]. \end{aligned} \quad (4)$$

By adjusting X , FDA estimates the causal effects as follows:

$$P(Y | do(X)) = \sum_m P(M = m | X) \sum_x P(x) [P(Y | M = m, X = x)]. \quad (5)$$

For brevity, we omit the transition details which share similar spirits to Equation (2). In essence, after adjustment, M becomes independent of X and $P(X = x | do(M = m)) = P(X = x)$, thus blocking the path $Y \leftarrow U \rightarrow X \rightarrow M$. More details of FDA can be found in [10]. To use FDA, a proper M should satisfy the following three requirements:

- (i) M intercepts all directed paths from X to Y ;
- (ii) there is no unblocked back-door path from X to M ;
- (iii) all back-door paths from M to Y are blocked by X .

3 METHOD

3.1 Problem Formulation

Recommendation. A recommendation dataset \mathcal{D} contains user-item interaction records $\{(x_i, y_i, o_i)\}_{i=1, \dots, |\mathcal{D}|}$ logged in a recommender system. x_i denotes the user data such as user ID and user historical behaviors. y_i denotes the item data such as item ID. $o_i \in \{0, 1\}$ is a binary factor indicating whether the user has interacted with the item. Given user data x and item data y , a recommendation model is learned to predict how likely the user will interact with the item, and the prediction score is used for candidate item ranking. Top-ranked items will appear in the recommendation list and are sorted according to the ranking scores.

Performance Heterogeneity. We define the total performance heterogeneity as the standard deviation of recommendation performances on different user groups' datasets $\{\mathcal{D}_e\}_{e \in \mathcal{E}}$ under *unified training*, i.e.,

$$S_h = \sqrt{\frac{1}{|\mathcal{E}| - 1} \sum_{e \in \mathcal{E}} (S(\mathcal{D}_e) - \bar{S})^2}, \quad (6)$$

where $S(\mathcal{D}^e)$ denotes the recommendation performance on the \mathcal{D}^e dataset and \bar{S} denotes the average performance on all datasets. Such performance heterogeneity includes the natural heterogeneity due to data imbalance across user groups, and the model bias that amplifies the natural heterogeneity and hurts the performance of unprivileged user groups. We define the natural performance heterogeneity as the standard deviation of recommendation performances on different user groups' datasets under *group-wise training*. Specifically, we independently train and evaluate one model per group where the performance is denoted as $S^*(\mathcal{D}_e)$. The natural performance heterogeneity can be written as:

$$S_h^* = \sqrt{\frac{1}{|\mathcal{E}| - 1} \sum_{e \in \mathcal{E}} (S^*(\mathcal{D}_e) - \bar{S}^*)^2}, \quad (7)$$

We define the performance heterogeneity brought by the model bias, i.e., bias-amplified performance heterogeneity as:

$$S_h^\circ = S_h - S_h^*. \quad (8)$$

In this paper, we use capital letters (e.g., X) to denote variables, lowercase letters (e.g., x) to denote particular values of variables, and bold ones (e.g., \mathbf{x}) to denote the vectorial representations.

3.2 Front-door Adjustment in Recommendation

In this subsection, we analyze the critical challenges in satisfying the three requirements of FDA and implementing FDA in deep recommendation models. Recall that in recommendation, X refers to the historical behaviors while Y refers to the next behavior (c.f. Figure 3). Many recommendation models follow the encoder-predictor structure where the encoder transforms X into latent factors, and the predictor determines the correlations between X and Y based on the latent factors and the target-item feature. As such, it is intuitive to view the latent factors obtained from X as the mediator M , and the representation of latent factors as the mediator feature. We analyze whether such M satisfies the requirements one by one:

Requirement (i). M should intercept all directed paths from X to Y . This means that there is no other information of X that could contribute to the recommendation prediction but is not encoded in M . Intuitively, if this requirement is not satisfied, there are other direct or indirect effect of X on Y besides $X \rightarrow M \rightarrow Y$, e.g., $X \rightarrow M^* \rightarrow Y$ or $X \rightarrow Y$. Then, front-door adjustment over each path (e.g., cutting $X \rightarrow M$ for $X \rightarrow M \rightarrow Y$ in step 2) would leave other paths $X \rightarrow M^* \rightarrow Y$ or $X \rightarrow Y$ still confounded by U , thus failing to estimate the causal effects. In the encoder-predictor recommendation architecture, the latent factors M as a whole contain all the information of the historical behaviors X which are used to make predictions. As such, the latent factors M as a whole obtained from the encoder satisfy the requirement.

Requirement (ii) & (iii). These requirements indicate that there are no confounders affecting (X, M) and no confounders other than U affecting (M, Y) , which are mostly satisfied if we use the latent factors obtained by the encoder as the mediator. Specifically, the parameters of the encoder and the predictor are randomly initialized, thus containing no confounder-related information before training. During training, the parameters are learned from (X, Y) pairs from the recommendation datasets, and can only encode confounders that affect (X, Y) , which are all included in U of the back-door path $X \leftarrow U \rightarrow Y$. As such, there are no confounders other than U affecting (M, Y) . (X, M) are not affected by the back-door path $X \leftarrow U \rightarrow Y \leftarrow M$ thanks to the colliding effect (c.f. Section 2.2). In addition, we show that we can improve the satisfaction of Requirement (ii) with causal feature distillation in Section 3.3.3.

With the mediator, Equation (5) is an expectation over the conditional distribution $P(m | X)$, and the prior distribution $P(x)$. First, we should sample different values m of M given X while a deterministic encoder can only output one m given user data x . Therefore, the sampling of M given X remains a challenge (**Challenge 1**). Second, the sample spaces of these two distributions are infinite, which makes the expectation calculation intractable. Moreover, the calculation of $P(Y | m, x)$ for each m and x is computationally expensive. Therefore, how to derive a proper approximation for expectation calculation, while preserving the **inference efficiency** due to the latency restrictions in recommender systems, presents the second challenge (**Challenge 2**).

3.3 Causal Multi-teacher Distillation

The ultimate goal of our method is to learn a recommendation model that could efficiently estimate the causal effect of user historical behaviors on the next behavior by handling unobserved confounders with FDA. The implementation of FDA faces the above two challenges in recommendation. To address **Challenge 1**, we propose to incorporate heterogeneous recommendation models where each model yields one mediator value m . As for **challenge 2**, we approximate the computationally expensive expectation calculation (*c.f.* Equation (5)), which enumerates each sampled m and x with knowledge distillation. That is, we first estimates the causal effects using heterogeneous teacher models (computationally expensive), and transfer the knowledge into an efficient student model (computationally light-weight). The student directly estimates the effect while trying to approaching the causal effect estimated by teachers via distillation losses. By unifying heterogeneous teachers and knowledge distillation, we propose the causal multi-teacher distillation framework, namely *CausalID*. In the following, we dive into the technical details of constructing heterogeneous teachers, performing FDA with heterogeneous teachers, and how we define the distillation losses. Key notations are summarized in Table 1.

3.3.1 Constructing Heterogeneous Teachers

We can split training recommendation data samples \mathcal{D} into non-IID subsets $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_K\}$ via biased selection *w.r.t.* some user attributes or unsupervised clustering. For example, we can select data samples linked with male and female users into two subsets, which have different data distributions due to the interest discrepancy between male and female users. In an extreme case, the user attribute is the user ID where each subset corresponds to a particular user. In practice, we choose the attribute related to an observed confounder of interest, such as user activeness or user behavior consistency. We illustrate that such a split could improve feature distillation in Section 3.3.3. We train one teacher model per subset, resulting in K heterogeneous teacher models that encode different data information. Each teacher contains an encoder, which extracts latent factors (mediator) from user historical behaviors, and a predictor, which makes recommendation predictions based on the latent factors and the target-item feature. We denote the encoders of pre-trained heterogeneous teacher models as $\Phi = \{\phi_k\}_{k=1,\dots,K}$, which will be used for obtaining mediator values. In the experiments (*c.f.* Section 5.4), we show that the *CausalID* framework significantly benefits more from heterogeneous teachers than homogeneous teachers trained on IID subsets, validating the rationality of constructing heterogeneous teachers for sampling mediator values.

The model architecture of each teacher remains the same as the student model. Therefore, the models are *unprivileged* teachers *w.r.t.* the student. A teacher is unprivileged if it has the same model architecture as the student and is trained on a subset of the dataset used to train the student. In other words, unprivileged teachers will bring no additional dataset knowledge nor model structure knowledge during distillation. This design permits a fair comparison with baseline models trained without distillation. In practice,

incorporating additional features or higher-capacity model architectures are acceptable for FDA and might further improve the recommendation performance, which is beyond the scope of this paper.

3.3.2 Front-door Adjustment for Causal Label Distillation

To estimate the causal effect with FDA, we sample multiple mediator values m with heterogeneous teachers and multiple x using the in-batch sampling strategy. We then calculate the causal effect as the expectation of $P(Y | M = m, X = x)$, which is modeled as a neural network, over sampled m and x (*c.f.* Equation (5)). The causal effect is the distillation label for training the student recommendation model. Specifically, the expectation calculation in Equation (5) can be decomposed into the following sub-processes:

Sampling of $M = m$. We use heterogeneous teacher recommendation models to do the sampling M given X . Given input x_i , the encoders $\Phi = \{\phi_k\}_{k=1,\dots,K}$ of heterogeneous teacher models yield heterogeneous $M = m$:

$$\tilde{\mathbf{m}}_{k,i} = \phi_k(x_i), \quad k = 1, \dots, K, \quad (9)$$

where $\tilde{\mathbf{m}}_{k,i}$ is the feature of mediator value $m_{k,i}$ encoded by ϕ_k given the input $X = x_i$. As for the modeling of $P(M = m_{k,i} | X)$, we take a parametric solution, *i.e.*, attention mechanism, and view the attention weights of X on multiple m as the conditional probability.

$$\alpha_{k,i} = \frac{\exp(\mathbf{W}_1 \tilde{\mathbf{m}}_{k,i} \cdot \mathbf{W}_2 \hat{\mathbf{m}}_i)}{\sum_k \exp(\mathbf{W}_1 \tilde{\mathbf{m}}_{k,i} \cdot \mathbf{W}_2 \hat{\mathbf{m}}_i)}, \quad \hat{\mathbf{m}}_i = \phi^*(x_i), \quad (10)$$

where $\tilde{\mathbf{m}}_{k,i}$ is the mediator feature for the i -th data extracted by the k -th teacher model and is computed by Equation (9). $\hat{\mathbf{m}}_i$ is the mediator feature of the i -th data computed by the encoder of the student model, *i.e.*, ϕ^* .

Sampling of $X = x$. We take the in-batch sampling strategy for efficiency. Specifically, for a given training sample x_i , we take other training samples $\{x_j\}_{j \neq i}$ in the batch as the sampled X . For the prior probability of $X = x_j$, we have no access to the data generation process in the real world. Alternatively, since each sample is randomly taken into the batch, we take the uniform distribution for $P(X = x_j)$.

Modeling of $P(Y | M = m_{k,i}, X = x_j)$. In recommender systems, the final prediction is generally pair-wise matching. For a user input x_i and a target item $Y = y_i$, the recommendation prediction is mostly a binary classification where the output \hat{o}_i indicates whether the user will interact with the item (*e.g.*, click or purchase). Therefore, we can parameterize $P(Y = y | M = m_{k,i}, X = x_j)$ as a neural network $\varphi(\cdot)$ followed by a sigmoid layer σ :

$$P(Y = y_i | X = x_j, M = m_{k,i}) = \sigma(\varphi(x_j, \tilde{\mathbf{m}}_{k,i}, \mathbf{y}_i)), \quad (11)$$

where $\tilde{\mathbf{m}}_{k,i}$ and \mathbf{y}_i are the vectorial representations of $m_{k,i}$ and y_i . $\tilde{\mathbf{m}}_{k,i}$ is extracted by the encoder of the k -th teacher recommendation model. In recommendation models, $\tilde{\mathbf{m}}_{k,i}$ can be the user representation extracted from the i -th user input data x_i (*e.g.*, user historical behavior sequence), and \mathbf{y}_i can be the embedding of item y_i obtained from a trainable embedding look-up table. Since M intercepts all the direct

TABLE 1: Notations

Notation	Description
X, Y, Z, M	variables denoting the historical behaviors, next behavior, confounder, mediator
x, y, z, m	particular values of X, Y, Z, M
$\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{m}$	vectorial representation of x, y, z, m
ϕ_k, ϕ^*	encoder of the k -th teacher / student recommendation model
$\tilde{\mathbf{m}}_{k,i}, \hat{\mathbf{m}}_i$	mediator feature extracted by ϕ_k and ϕ^* that take x_i as input
$\tilde{\mathbf{m}}_i^\circ$	mediator feature estimated by teachers with back-door adjustment
o, \tilde{o}, \hat{o}	recommendation label (e.g., CTR), prediction estimated by teachers with front-door adjustment, prediction made by the student
ψ_{FDA}	fully-connected layers for estimating \tilde{o} with front-door adjustment
ψ^*	fully-connected layers for estimating \hat{o}
$\mathbf{W}_1, \mathbf{W}_2$	trainable matrices for estimating the conditional probability $P(M = m X)$
\mathcal{L}_{BDA}	feature distillation loss with back-door adjustment
\mathcal{L}_{FDA}	label distillation loss with front-door adjustment
\mathcal{L}_{Rec}	original recommendation loss (e.g., BPR loss, Sampled Softmax loss)
$\lambda_{BDA}, \lambda_{FDA}$	coefficients to control $\mathcal{L}_{BDA}, \mathcal{L}_{FDA}$

effects from $X \rightarrow Y$, we can use the mediator m_j to replace x_j and use the vectorial representation $\tilde{\mathbf{m}}_j$ in the modeling:

$$P(Y = y_i | X = x_j, M = m_{k,i}) = \sigma(\varphi(\tilde{\mathbf{m}}_j, \tilde{\mathbf{m}}_{k,i}, \mathbf{y}_i)), \quad (12)$$

where $\tilde{\mathbf{m}}_j = 1/K \sum_k \tilde{\mathbf{m}}_{k,j}$ is the mediator feature of data x_j by taking the average of the mediator features of all teachers. Function φ can be modeled by any neural network architecture that makes recommendation predictions. Taking DIN [1] as an example, DIN uses fully-connected layers (FCL) to make click-through-rate predictions with concatenated user-item representations as input. Following this work, we use a FCL network ψ_{FDA} to model φ :

$$\varphi(\tilde{\mathbf{m}}_j, \tilde{\mathbf{m}}, \mathbf{y}) = \psi_{FDA}([\tilde{\mathbf{m}}_j, \tilde{\mathbf{m}}_{k,i}, \mathbf{y}_i]), \quad (13)$$

where $[\cdot]$ denotes vector concatenation operation.

Estimating $P(Y|do(X))$. With the sampling of $M = m$ (cf. Equation (9) - (10)), the sampling of $X = x$, the modeling of $P(Y | M = m_{k,i}, X = x_j)$ (cf. Equation (11) - Equation (13)), we can estimate the causal effect of user data x_i on the next item y_i by implementing FDA (cf. Equation (5)) as follows:

$$\tilde{o}_i = \sum_k \alpha_{k,i} \sum_j \frac{1}{N_b} \sigma(\psi_{FDA}([\tilde{\mathbf{m}}_j, \tilde{\mathbf{m}}_{k,i}, \mathbf{y}_i])), \quad (14)$$

where $\alpha_{k,i}$ is computed with Equation (10), and \tilde{o}_i is the causal recommendation prediction indicating the probability of the interaction between x_i and y_i .

Causal Label Distillation. \tilde{o}_i is predicted by all teacher recommendation models, and serves as the **causal label guidance** for training the student recommendation model. The prediction of the student model is obtained as follows:

$$\hat{o}_i = \psi^*([\hat{\mathbf{m}}_i, \mathbf{y}_i]), \quad (15)$$

where ψ^* denotes the predictor of the student recommendation model, which is modeled as another FCL network. $\hat{\mathbf{m}}_i$ is the mediator feature extracted by the student encoder ϕ^* . Since the Equation (14) introduces trainable parameters i.e., $\mathbf{W}_1, \mathbf{W}_2$, and ϕ , we try to pull the prediction closer to the ground-truth label o_i . Then, the loss function of FDA can be:

$$\mathcal{L}_{FDA} = \sum_i (\underbrace{L_{kd}(\tilde{o}_i, \hat{o}_i)}_{\text{Distillation}} + \underbrace{L_{ce}(\tilde{o}_i, o_i)}_{\text{Consistency}}), \quad (16)$$

where L_{kd} is the knowledge distillation criterion [21]. L_{ce} is the cross-entropy loss. It is noteworthy that the consistency loss solely affects $\mathbf{W}_1, \mathbf{W}_2$, and ψ_{FDA} . The distillation loss is computed when $\mathbf{W}_1, \mathbf{W}_2$, and ψ_{FDA} are fixed.

3.3.3 Feature Distillation

Besides label distillation, we also incorporate the feature distillation loss for training the student recommendation model. Note that the mediator values m extracted by teachers are intermediate features, which can serve as the feature guidance. The feature distillation loss can be defined as:

$$\mathcal{L}_{FT} = \sum_i d(\pi(\{\tilde{\mathbf{m}}_{k,i}\}_{k=1}^K), \hat{\mathbf{m}}_i), \quad (17)$$

where π denotes the function that aggregates the intermediate features of teachers. d denotes a distance function such as Euclidean distance. π is decided according to whether we can observe the confounder of interest. That is, if the confounder of interest is not observed, we take the vanilla feature distillation. If the confounder of interest is observed, we employ back-door adjustment to improve deconfounding in the feature level, namely causal feature distillation.

Vanilla Feature Distillation. The aggregation function in the vanilla feature distillation takes the average of all teacher features, which can be written as follows:

$$\pi(\{\tilde{\mathbf{m}}_{k,i}\}_{k=1}^K) = \frac{1}{K} \sum_k \tilde{\mathbf{m}}_{k,i}. \quad (18)$$

Causal Feature Distillation. Besides the vanilla feature distillation, we propose an optional causal feature distillation to handle some confounders Z of interest that can be observed in recommender systems with back-door adjustment (c.f. Section 2.1). First, we can split the data w.r.t. different confounder values. By taking user activeness as an example, we can count the number of interactions and accordingly split users into several groups. Since data samples in the k -th group are selected when the confounder variable is set to a particular value $Z = z_k$, the parameters of the corresponding models learn the knowledge under $Z = z_k$. Then, teachers can be regarded as the estimators $P(M | X, Z = z_k)$, and together approximate $P(M | do(X))$ as the weighted

sum of the estimation of different models according to Equation (2). Formally, we have:

$$\tilde{\mathbf{m}}_i^\circ = \mathbb{E}_{z_k} [\phi_k(x_i)] = \sum_k \tilde{\mathbf{m}}_{k,i} P(z_k). \quad (19)$$

We set $P(z_k)$ to be proportional to the number of users of the group where ϕ_k is trained. We take $\tilde{\mathbf{m}}_i^\circ$ as the **causal feature guidance** for training the student as follows:

$$\mathcal{L}_{BDA} = \sum_i d(\tilde{\mathbf{m}}_i^\circ, \hat{\mathbf{m}}_i). \quad (20)$$

This modeling can better satisfy Requirement (ii) in FDA by actively blocking back-door paths from X to M , to meet the complex environments in real-world recommender systems. We also replace the average mediator feature $\tilde{\mathbf{m}}_j$ in Equation (13) - (14) with $\tilde{\mathbf{m}}_j^\circ$ to improve deconfounding since $\tilde{\mathbf{m}}_j^\circ$ represents the debiased mediator feature.

3.4 Model Training

In summary, we first split the recommendation dataset \mathcal{D} into non-IID subsets $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_K\}$ with biased section w.r.t. some user attributes. If we aim to address the model bias caused by a particular confounder, we can split the data according to the confounder values. We train one recommendation model per subset as one teacher, and obtain their encoders $\{\phi_1, \phi_2, \dots, \phi_K\}$. Given (x_i, y_i, o_i) , we first obtain causal feature guidance $\tilde{\mathbf{m}}_i^\circ$ from all teachers with Equation (19), and the mediator feature $\hat{\mathbf{m}}_i$ extracted by the student encoder ψ^* . As such, the **causal feature distillation** loss \mathcal{L}_{BDA} is computed with Equation (20). We then obtain the causal label guidance \tilde{o}_i from all teachers with Equation (14), and the prediction \hat{o}_i of the student recommendation model with Equation (15). The **label distillation** loss and the consistency loss are given in Equation (16). Besides two distillation losses, we also employ a widely used recommendation objective \mathcal{L}_{Rec} , such as BPR loss [22] or the sampled softmax loss [23], to train the student recommendation model, e.g.,

$$\mathcal{L}_{Rec} = \sum_{(x_i, y_i, o_i) \in \mathcal{D}} -\ln \sigma(\hat{o}_i - o_i) + \lambda \|\Theta\|_2^2, \quad (21)$$

where \hat{o}_i denotes the prediction of the student model. \hat{o}_i is basically $P(Y | X)$ but approximates $P(Y | do(X))$ with causal label distillation (*cf.* Section 3.3.2). During inference, we use the prediction \hat{o}_i of the student model for recommendation. Overall, the training objective can be written as:

$$\mathcal{L}_{CausalD} = \mathcal{L}_{Rec} + \lambda_{BDA} \mathcal{L}_{BDA} + \lambda_{FDA} \mathcal{L}_{FDA}. \quad (22)$$

The training algorithm of the causal multi-teacher distillation is summarized in Algorithm 1.

3.5 Method Analysis

3.5.1 Training Complexity

The computation complexity of CausalD mainly comes from the following processes:

- **Teacher Pre-training.** Teachers are trained on non-overlapping subsets of the whole recommendation dataset. Each teacher is with the same neural network

Algorithm 1: Causal Multi-teacher Distillation

```

Input: Recommendation data  $\mathcal{D}$ 
Output: Student recommendation model  $\{\phi^*, \psi^*\}$ 
Split  $\mathcal{D}$  into non-IID subsets  $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_K\}$ 
for ( $k = 1$  to  $K$ ) {
    Train one teacher on  $\mathcal{D}_k$  and obtain encoder  $\phi_k$ 
}
Initialize  $\Theta = \{\phi^*, \psi^*, \mathbf{W}_1, \mathbf{W}_2, \psi_{FDA}\}$ 
while not converged do
    Sample a batch  $\{(x_i, y_i, o_i)\}_{i=1, \dots, N_b}$  from  $\mathcal{D}$ 
    for ( $i = 1$  to  $N_b$ ) {
         $\hat{\mathbf{m}}_i = \phi^*(x_i)$ 
         $\tilde{\mathbf{m}}_i^\circ = \mathbb{E}_{z_k} [\phi_k(x_i)]$   $\triangleright$  Eq. (19)
        Obtain  $\tilde{o}_i$  with Eq. (14)
        Obtain  $\hat{o}_i$  with Eq. (15)
    }
     $\mathcal{L}_{BDA} = \sum_i d(\tilde{\mathbf{m}}_i^\circ, \hat{\mathbf{m}}_i)$   $\triangleright$  Eq. (20)
     $\mathcal{L}_{FDA} = \sum_i (L_{kd}(\tilde{o}_i, \hat{o}_i) + L_{ce}(\tilde{o}_i, o_i))$   $\triangleright$  Eq. (16)
    Obtain  $\mathcal{L}_{Rec}$  with BPR [22], etc.
    Optimize  $\Theta$  over Eq. (22)
end

```

architecture as the student. As such, supposing the number of epochs in training the teachers and the student remains the same, the complexity of training all teachers approximates the complexity of training the student.

- **Feature encoding.** Supposing that the encoding complexity of the student is $O(E)$, the encoding complexity of the causal distillation framework will be $O(KE + E)$ where K denotes the number of teachers. Note that the teacher encoder and the student encoder have the same structure.
- **Prediction.** Supposing that there are L^* layers in ψ^* and the number of hidden units in the l -th layer is d_l^* , the prediction complexity of the student can be $O(\sum_{l=1}^{L^*} d_l^* d_{l-1}^*)$. Supposing that there are \tilde{L} layers in ψ_{FDA} and the number of hidden units in the l -th layer is \tilde{d}_l , the prediction complexity of the teachers can be $O(\sum_{l=1}^{\tilde{L}} N_b K \tilde{d}_l \tilde{d}_{l-1})$.

In summary, the computation complexity in causal distillation training can be $O(KE + E + 2 \sum_{l=1}^{\tilde{L}} N_b K \tilde{d}_l \tilde{d}_{l-1} + \sum_{l=1}^{L^*} d_l^* d_{l-1}^*)$. In practice, N_b samples can be processed in parallel, and $\sum_{l=1}^{\tilde{L}} N_b K \tilde{d}_l \tilde{d}_{l-1}$ introduces acceptable complexity when \tilde{d}_l and \tilde{L} are small, i.e., ψ_{FDA} is a light-weight network compared to ψ^* . Empirically, with DIN [1] as the Base model, the multi-teacher pre-training costs around 72s per epoch on the MovieLens dataset. The causal multi-teacher distillation training costs around 157s per epoch. The Base model without distillation costs 64s per epoch under the same experimental setting.

3.5.2 Inference Complexity

At inference, solely the student recommendation model is used, and all teachers are discarded. Items are ranked according to the student predictions \hat{o} . Therefore, the inference complexity remains the same with the Base model without distillation. Note that the Base model can be any recommendation model with an encoder-predictor architecture. In other words, the causal distillation framework can

be plugged into existing recommendation models without affecting the inference efficiency.

3.5.3 Connection between causal label distillation and causal feature distillation

The underlying connections between causal label distillation and causal feature distillation can be two-fold:

- Front-door adjustment and back-door adjustment are two representative causal techniques for deconfounding with unobserved and observed confounders, respectively. We instantiate these two techniques in one unified neural model by identifying mediators and observed confounders. We empirically find that the joint modeling contributes to the overall effectiveness.
- As described in Section 3.2, the requirement (ii) of applying front-door adjustment is that no confounders simultaneously affect X and M . We further apply the back-door adjustment for deconfounding the estimation of causal effects of X on M (cf. Figure 3 Back-door Adjustment).

4 RELATED WORKS

4.1 Debiased Recommendation

Biases are ubiquitous in the feedback loop (User → Data → Model → User) of recommender systems, including (but are not limited to) exposure bias [24], [25], popularity bias [26], conformity bias [27], position bias [28], and selection bias [29]. One representative line of research to alleviate these biases is to explicitly model or utilize these biases [30]. Typically, they construct auxiliary models like exposure models [25], [31], click models [32], and temporal popularity models [33]. Another representative direction is to leverage unbiased data [19], [34]. For example, [19] collect uniform data by exposing randomly sampled items to each user with a uniform distribution. Undoubtedly, random exposure might hurt users' experiences and the uniform data is expensive to collect. More recently, there is substantial and rapidly-growing research literature studying embracing causal theory in bias alleviation [16], [18], [31], [35], [36], [37], [38]. Among these, Inverse Propensity Score [14], [39] is one of the most general and representative methods in reducing confounding effects. In essence, IPS re-weights training samples based on the estimated bias-aware propensity scores. However, most existing deconfounding works (including IPS) assume unconfoundedness [20], which means the confounders are observable to the model. However, in real-world recommender systems, confounders are too various to be modeled in one model, and a considerable portion of them are even unobserved by the system. Different from these works, we leverage front-door adjustment technique from causality to handle unobserved confounders, and derive an efficient approximation of it using causal multi-teacher distillation.

It is noteworthy that we differ from fairness-related works [9] both in motivation and technique. Our major focus is the model training bias that enlarges the performance heterogeneity of different user groups. As shown in Figure 1, when users in different groups are trained all together, and trained independently, a small portion of users benefit

TABLE 2: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	#Density
MovieLens	6,040	3,900	1,000,209	0.04246
Amazon	459,133	313,966	8,898,041	0.00063
Alipay	400,594	19,503	38,710,494	0.00495

more during the co-training of all users. Note that performance heterogeneity still exists without co-training, and such heterogeneity is *not* our major focus. Fairness-related works typically penalize the prediction expectation (such as the positive prediction rate) of different user groups, ensuring the universal equality. For example, [9] work in a post-processing manner with recommended Top-K items and their ranking scores as input. They try to ensure equal performance of different user groups by solving a 0-1 integer programming problem. Different from existing works, we use causal techniques to deconfound model training.

4.2 Knowledge Distillation

Multi-teacher Distillation. Knowledge distillation is a powerful tool for transferring knowledge between networks. The transferred knowledge takes the form of soft labels [21], intermediate features [40], [41], and relations [42], [43]. More recently, there is a rapidly-growing research literature studying on multi-teacher distillation [44], [45]. Among them, MEAL [44] is one of the early works that train multiple teachers. It measures the consistency between the selected teacher and the student using adversarial learning. [45] encapsulates multi-teacher distillation into multi-task learning, and trains one teacher per task. EnsCTR [46] makes the first attempt to ensemble multiple teachers for click-through-rate prediction. They devise the Teacher Gating network to assign weights to different teachers based on their predictions. To the best of our knowledge, our work is the first to consolidate causality into multi-teacher distillation.

Knowledge Distillation in Recommendation. In recommender systems, there are some works that exploit a privileged teacher to enhance the student that is used for serving/inference [47], [48]. The privilege of teachers mainly falls into two groups: 1) high-capacity in model's architecture [49], [50] where related works typically train a lighter-weight student for efficiency; and 2) external knowledge such as user reviews [51], on-device features [52], and unbiased uniform data [19]. Largely different from these works, we construct unprivileged teachers, which have the same architecture as the student, and each of them is solely on a subset of the training data used to train the student. Though these teachers are unprivileged, we distill causal effects from them and accordingly train a debiased student without external unbiased data, which constitutes a novel paradigm for leveraging distillation in recommender systems.

5 EXPERIMENTS

We conduct experiments on three real-world datasets to answer the following research questions:

- **RQ1:** How does *CausalD* perform as compared to baselines concerning multi-teacher distillation, debiased distillation, and debiased recommendation?

TABLE 3: Overall performance and performance heterogeneity comparison with DIN as the base model. p -value is from the two-sided test between CausalD and the best-performing baseline. Higher values are better except for Heterogeneity.

Datasets	Metric	DIN	KD	IPS	CausalRec	DeRec	DebiasD	EnsCTR	MEAL	CausalD	p -value
MovieLens	AUC	0.8185	0.8187	0.8034	0.8176	0.8227	0.8227	<u>0.8257</u>	0.8233	0.8329	1.36e-07
	R@5	0.3102	0.3104	0.3066	0.3112	0.3260	0.3231	<u>0.3354</u>	0.3213	0.3556	3.55e-06
	R@10	0.4682	0.4763	0.4652	0.4751	0.4892	0.4831	<u>0.4930</u>	0.4837	0.5170	4.67e-05
	NDCG@5	0.2378	0.2382	0.2362	0.2400	0.2507	0.2485	<u>0.2589</u>	0.2492	0.2755	1.15e-06
	NDCG@10	0.3145	0.3187	0.3136	0.3192	0.3300	0.3258	<u>0.3355</u>	0.3281	0.3538	5.53e-06
	Heterogeneity ↓	0.7291	0.4533	<u>0.3056</u>	0.7529	0.7893	0.6889	0.9531	0.7773	0.1086	1.54e-03
Amazon	AUC	0.8873	0.8704	0.8788	0.8899	0.8908	0.8954	<u>0.8983</u>	0.8867	0.9027	2.30e-07
	R@5	0.5644	0.5336	0.5555	0.5790	0.5795	0.5886	<u>0.5911</u>	0.5698	0.6069	3.24e-04
	R@10	0.6854	0.6515	0.6760	0.6956	0.6956	0.7060	<u>0.7087</u>	0.6863	0.7216	1.52e-05
	NDCG@5	0.4794	0.4516	0.4706	0.4934	0.4939	0.5007	<u>0.5038</u>	0.4857	0.5190	3.71e-05
	NDCG@10	0.5384	0.5092	0.5294	0.5503	0.5506	0.5580	<u>0.5612</u>	0.5426	0.5751	2.58e-05
	Heterogeneity ↓	2.4567	1.6847	<u>1.4277</u>	1.8765	1.6241	1.4747	1.5169	2.2083	1.3803	1.89e-02
Alipay	AUC	0.7691	0.7615	0.7623	0.7603	0.7738	0.7712	<u>0.7700</u>	<u>0.7749</u>	0.7777	3.08e-05
	R@5	0.1669	0.1727	0.1778	0.1792	0.2039	0.2057	0.1831	<u>0.2343</u>	0.2547	5.73e-08
	R@10	0.3518	0.3682	<u>0.3953</u>	0.3772	0.3947	0.3675	0.3689	0.3938	0.4457	2.22e-06
	NDCG@5	0.1186	0.1221	0.1237	0.1255	0.1474	0.1532	0.1307	<u>0.1745</u>	0.1851	1.53e-06
	NDCG@10	0.2076	0.2165	0.2289	0.2214	0.2407	0.2314	0.2204	<u>0.2517</u>	0.2779	1.86e-05
	Heterogeneity ↓	2.3419	2.3383	1.8455	<u>1.5405</u>	2.4671	2.1373	2.4441	2.1125	1.0207	3.39e-03

- RQ2:** Does *CausalD* mitigate the performance heterogeneity problem amplified by confounders that are observed or unobserved by the model?
- RQ3:** How do different components (the back-door and front-door adjustment loss) and different hyper-parameter settings (e.g., loss coefficients, number of teachers) affect the performance of *CausalD*?

5.1 Experimental Setup

Dataset Description. We use two public benchmark datasets, *i.e.*, MovieLens and Amazon Product Review, and a challenging large-scale industrial dataset, *i.e.*, AliPay. Dataset statistics are summarized in Table 2.

- Amazon.** We take the Book category from the Amazon product review dataset⁴ for evaluation. We follow [1], [53] to regard the interacted items for a given user as positive items and pair the user with randomly sampled negatives that the user did not consume before for training models.
- MovieLens.** We use the widely used MovieLens-1M⁵ version for evaluation. We generate positive and negative pairs in a similar manner as the Amazon dataset.
- AliPay.** AliPay dataset is collected from one of the world's largest mobile payment platforms, *i.e.*, AliPay. Wherein, the applets like mobile recharge services and government services are viewed as items. In AliPay, we regard the items that have been clicked as positives. Items that are observed but not clicked are treated as negatives.

To ensure data quality, we take the 16-core setting where each user/item has at least sixteen interactions.

Baselines. To demonstrate the effectiveness, we compare *CausalD* with the following baselines:

- KD [21]:** This model denotes the oracle knowledge distillation. We train a single teacher on all users.

⁴<http://jmcauley.ucsd.edu/data/amazon/>

⁵<https://grouplens.org/datasets/movielens/1m/>

- IPS [39]:** It is a representative causal recommendation method. In this work, we firstly measure $P(ug|i)$ which denotes the conditional probability of heterogeneous user groups ug given items i . Then, for a given data sample (u, i) , we use the corresponding $p(ug|i)$ as the propensity score. We follow [39] to reduce propensity variance using the propensity clipping technique. The clipping threshold is searched in $\{1/2, 1/3, \dots, 1/10\}$.
- CausalRec [54]:** This method employs causal inference to correct the exposure bias and technically constructs an exposure model such that each click is weighted by the inverse exposure probability predicted by the model.
- Deconfounded Rec (DeRec) [14]:** The deconfounded recommendation technique uses Poisson factorization to infer confounders, and augments the base recommendation model to correct the confounding.
- DebiasD [55]:** It is one state-of-the-art method that debiases single-teacher distillation at the class level. We extend the framework to the multi-teacher distillation by assigning adaptive mixing weights to multiple teachers.
- EnsCTR [46]:** It is one state-of-the-art multi-teacher distillation framework in the literature of recommendation.
- MEAL [44]:** This baseline is a state-of-the-art ensemble distillation in the generic domain. It incorporates discriminators to evaluate to what extent the student's output differs from multiple teachers.

Implementation Details and Evaluation. Since *CausalD* is designed as model-agnostic, we instantiate *CausalD* and baselines on two representative base recommenders, *i.e.*, Deep Interest Network (DIN) [1] and DeepFM [56]. All models except for KD, which is a single-teacher distillation baseline, share the same pre-trained multiple teachers. We optimize all models with Adagrad [57] as the optimizer and with the default choice of learning rate (*i.e.*, 0.01). All models are with batch size 4096, and user/item embedding size 8. For DIN and models constructed upon it, we consider the last 50 items interacted to form the user sequence.

TABLE 4: Recommendation performance and performance heterogeneity comparison with DeepFM as the base model.

Datasets	Metric	DeepFM	KD	IPS	CausalRec	DeRec	DebiasD	EnsCTR	MEAL	CausalD	<i>p</i> -value
MovieLens	AUC	0.8142	0.8234	0.8113	0.8172	0.8142	0.8231	<u>0.8241</u>	0.8210	0.8320	4.68e-03
	R@5	0.2867	0.3047	0.2867	0.3021	0.2902	0.3064	<u>0.3142</u>	0.3039	0.3249	3.88e-02
	R@10	0.4463	0.4633	0.4469	0.4537	0.4509	0.4685	<u>0.4787</u>	0.4691	0.4916	2.66e-02
	NDCG@5	0.2230	0.2362	0.2188	0.2338	0.2241	0.2348	<u>0.2420</u>	0.2354	0.2481	1.47e-02
	NDCG@10	0.3005	0.3129	0.2967	0.3072	0.3019	0.3134	<u>0.3216</u>	0.3152	0.3290	2.62e-02
	Heterogeneity ↓	7.1753	5.8783	5.4983	5.7853	5.7893	6.4233	6.1513	5.9233	<u>5.6333</u>	-
Amazon	AUC	0.8941	0.8827	0.8886	0.8957	0.8956	0.8953	0.8913	<u>0.8963</u>	0.8977	3.39e-03
	R@5	0.5693	0.5723	0.5568	0.5730	0.5718	0.5743	<u>0.5855</u>	0.5760	0.6039	6.62e-06
	R@10	0.6877	0.6880	0.6775	0.6918	0.6904	0.6923	0.6934	<u>0.6947</u>	0.7087	1.89e-03
	NDCG@5	0.4838	0.4876	0.4721	0.4869	0.4865	0.4886	<u>0.5018</u>	0.4904	0.5191	1.49e-04
	NDCG@10	0.5416	0.5440	0.5310	0.5449	0.5444	0.5461	<u>0.5544</u>	0.5483	0.5716	6.94e-03
	Heterogeneity ↓	0.2224	<u>0.1422</u>	0.1082	0.1840	0.2350	0.1988	0.2376	0.1414	0.0762	5.10e-04
Alipay	AUC	0.8018	0.8006	<u>0.8058</u>	<u>0.8058</u>	0.8016	0.8017	0.8077	0.8016	0.8031	-
	R@5	0.2094	0.2227	0.2253	0.2326	0.2281	0.2274	<u>0.2612</u>	0.2292	0.2909	2.61e-04
	R@10	0.4177	0.4263	0.4276	0.4314	0.4298	0.4313	<u>0.4492</u>	0.4257	0.4657	2.16e-02
	NDCG@5	0.1491	0.1615	0.1626	0.1674	0.1633	0.1637	<u>0.1939</u>	0.1661	0.2207	1.63e-05
	NDCG@10	0.2501	0.2603	0.2607	0.2639	0.2614	0.2626	<u>0.2848</u>	0.2614	0.3053	1.06e-02
	Heterogeneity ↓	2.0041	2.0877	1.6609	1.6577	1.8715	1.6355	2.3579	<u>1.5945</u>	1.5747	2.31e-02

For loss coefficients of all models that introduce additional loss functions (including \mathcal{L}_{BDA} and \mathcal{L}_{FDA} in CausalD), we search in the range of $\{0.01, 0.1, 1, 10\}$. We run CausalD and the best-performing baseline for five times independently, and conduct two-sided tests.

It is known that industrial recommender systems typically consist of two phases, *i.e.*, the matching phase (also known as deep candidate generation) and the ranking phase. In the literature, evaluation strategies for these two lines of works are different. For example, ranking models solely sort a small group of candidates rather than the whole item gallery. Therefore, ranking models can incorporate complex feature interactions between items and users. In this paper, we focus on the ranking phase and obey the next-item recommendation protocol to evaluate the performance of models, which has been widely used in [1], [2]. Specifically, we chronologically order the interacted items for each user to have a behavior sequence, and leave the last element of the behavior sequence for testing. The testing target item is paired with 100 randomly sampled negative items that the user has not consumed. To have a comprehensive analysis of models, we employ various *evaluation metrics*, including Area Under the ROC Curve (AUC), Normalized Discounted Cumulative Gain (NDCG), and Recall. We report metrics computed on the top 5/10 items. Besides these widely used metrics, we also evaluate the performance heterogeneity of different users. The *Heterogeneity* metric is the mean of five bias-amplified performance heterogeneity scores computed according to Equation 8 over five metrics (AUC, R@5, R@10, NDCG@5, NDCG@10). The lower values, the better.

5.2 Overall Performance (RQ1)

We run CausalD and all baselines on three datasets. For each model, we run it five times and take the average averaged results. The results with DIN and DeepFM as base models are shown in Table 3 and Table 4, respectively. Higher values are better for AUC, NDCG, and Recall except for Heterogeneity. From the comprehensive results, we have the following observations:

- In summary, CausalD achieves the best performance and the least performance heterogeneity in most cases. For example, CausalD outperforms DIN and the best performing baseline EnsCTR by 15.85% and 6.41% (NDCG@5), and reduces the performance heterogeneity by 8.14% and 11.09% on the MovieLens dataset. The performance gains are statistically significant with p -value < 0.05 under two-sided tests⁶. Improving the overall performance while reducing the heterogeneity probably indicates that tail users benefit more with CausalD compared to other baselines. We further reveal the rationality of this analysis in Section 5.3. On the contrary, some other baselines (such as IPS) achieve low heterogeneity and low performance at the same time, which means that they might hurt the performance of head users. The improvements are consistent across different recommendation architectures and different datasets, showing the merits of CausalD being model-agnostic and domain-agnostic. On the larger-scale dataset, Alipay, the improvements are even more substantial. In essence, these results demonstrate the necessity of reducing confounding effects for recommender systems, and the effectiveness of the causal distillation framework.
- The oracle distillation framework (KD) obtains almost no gains *w.r.t.* the base models. This indicates that when the model capacity and the training data remain the same, the unprivileged teacher might not be very useful in recommender systems. This method can reduce the heterogeneity to a limited extent. We attribute this result to that soft labels produced by the teacher can probably convey some confidence judgments on the predictions for different users. For example, active users might get smoother soft labels than inactive ones.
- IPS reduces the performance heterogeneity in many cases by considering user-specific propensity scores. However, it obtains almost no gains on recommendation performance and sometimes does harm on MovieLens and Amazon with DIN as the base model. One reasonable

⁶https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

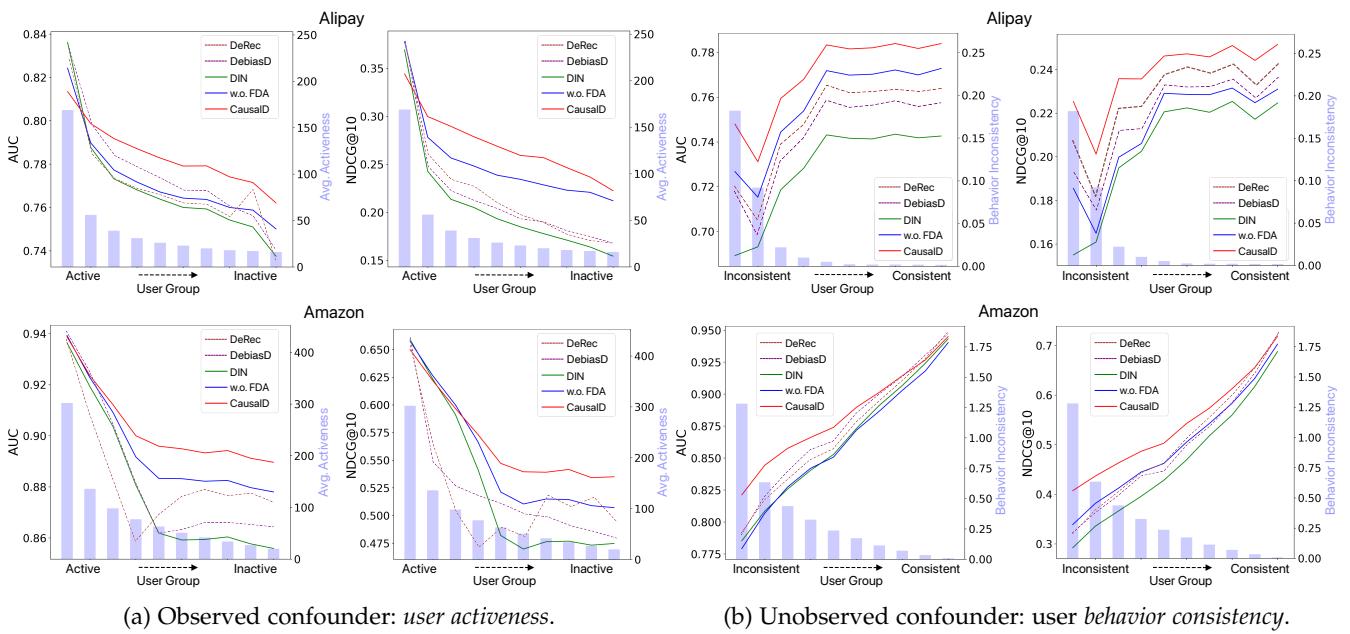


Fig. 4: Performance heterogeneity across different user groups *w.r.t.* two confounders, *i.e.*, user activeness, and behavior consistency. The performance heterogeneity over user groups is largely reduced compared to the base model DIN [1].

explanation is that IPS largely sacrifices the performance of head users for the performance of long-tail users. Compared to IPS, exposure-aware causal baselines CausalRec and DeRec can consistently improve the Base model. By correcting the exposure bias, these models are potentially less affected by the spurious correlations between users and frequently exposed items. Tail users could benefit from the correction since they suffer more from spurious correlations as illustrated in the Introduction. However, they might not effectively handle other ubiquitous confounders (*e.g.*, user activeness, item popularity) beyond exposure. The debiased distillation baseline DebiasD can also obtain some performance gains and reduces the performance heterogeneity to some extent. It learns whether we should trust teachers and how differently we should trust different teachers in distillation. But still, its results are inferior to CausalD in most cases.

- Muti-teacher distillation baseline EnsCTR achieves good performance compared to other baselines. It leverages multi-teachers to maximize the recommendation performance. However, the heterogeneity is mostly worse than other baselines and even worse than the base model in some cases, which probably means EnsCTR further sacrifices the performance of tail users. The performance of MEAL seems to be less consistent in different experimental settings. Unstable training caused by adversarial learning is also found in many other research fields [58].
- We also notice that with DeepFM as the base model, CausalD cannot beat other baselines in a few cases while CausalD with DIN as the base model could consistently achieve the best results. We attribute this phenomenon to the two parallel components, *i.e.*, the FM component and the DNN component, of DeepFM. On the one hand, the causal feature distillation that operates on intermediate representations is mostly affecting the DNN component

while the FM component might still suffer from training biases. On the other hand, two parallel components would result in two independent predictions while their relative importance might be hard to determine under causal label distillation. Still, we observe that CausalD over DeepFM could achieve the best results in most cases, demonstrating the model-agnostic effectiveness of CausalD.

5.3 Deconfounding Analysis (RQ2)

We are interested in whether CausalD reduces the performance heterogeneity of users by deconfounding and improving the performance of tail users with minor performance sacrifice of head users. In this regard, we report the recommendation performance of models on different user groups split *w.r.t.* two confounders, *i.e.*, 1) *user activeness*, which is observed by the model, and 2) users' *behavior consistency* on popular items, which is unobserved by the model. We evaluate CausalD, a variant of CausalD without loss \mathcal{L}_{FDA} (w.o. FDA), and the Base DIN model. Figure 4a and Figure 4b show the results. We plot both the recommendation performance (the lines) and the values of confounders (the bars), including user activeness and behavior inconsistency. According to the results, we have the following observations:

- Overall, these results further show the effectiveness of CausalD on improving recommendation performance and reducing performance heterogeneity of different users.
- By analyzing the performance over different groups in detail, we observe that CausalD improves the recommendation performance by mainly and largely improving the results of tail users that suffer from low-quality recommendation while mostly preserving the recommendation quality of head users. These results basically demonstrate that CausalD reaches the major goal of this work, *i.e.*, to alleviate model training bias that is in favor of head

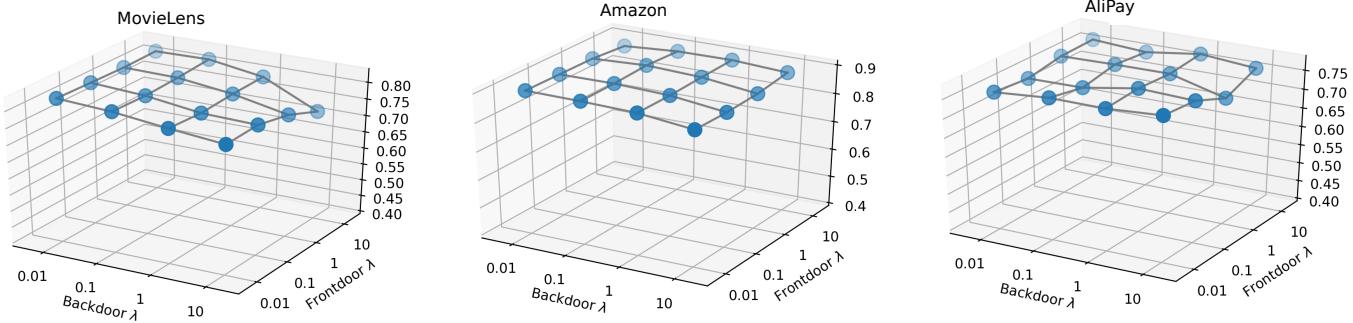


Fig. 5: Recommendation performance with loss coefficients (λ_{BDA} and λ_{FDA}) varying in range $\{0.01, 0.1, 1, 10\}$.

users, and to reduce performance heterogeneity by mainly improving the performance of tail users that originally benefit little from the co-training of all users. This merit can be potentially of great value in real-world recommender systems where tail users are the majority and cold-start problems are ubiquitous.

- Although the DeRec causal recommendation baseline and the DebiasD debiased distillation baseline can provide some relief for performance heterogeneity issues in certain scenarios, there remains a significant disparity in bias alleviation performance compared to causalD. These results are consistent with those outlined in Section 5.2 and the analytical observations was comparable.
- CausalD can successfully handle confounders (e.g., behavior consistency) that are unobserved to the model. This is technically critical merit of the CausalD framework. In real-world scenarios, we cannot observe all confounders that cause the spurious correlations of historical behaviors and the next behaviors, resulting in inevitable performance heterogeneity and non-robustness. Dealing with unobserved confounders is enticing but challenging, and this work demonstrates the front-door adjustment might be a promising technique for recommender systems.
- We also notice that, in some cases, the performance of CausalD on head users becomes worse than the Base model. This phenomenon can be intuitive since by alleviating training bias brought by potential confounders, the model might capture less biased prediction shortcuts that are in favor of the majority user group. For example, some active users might click a lot of popular items disregarding their inherent interests, leading to biased records. Debiased models might recommend items that are less affected by the item popularities, and thus probably hurting the performance of the above users. Still, CausalD achieves consistently better performance for all user groups in most cases. We leave user-wise bias exploitation as a potential future work.

esults of causal recommendation baseline DeRec and debiased distillation baseline DebiaseD in Figure 4

5.4 In-depth Model Analysis (RQ3)

Ablation Studies. We are interested in whether the critical components in CausalD all contribute to its effectiveness. In this regard, we evaluate whether 1) heterogeneous teachers outperform homogeneous teachers; and 2) whether the

TABLE 5: Ablation study with DIN as the base model *w.r.t.* three critical components: 1) front-door adjustment (w.o. FDA \rightarrow CausalD); 2) back-door adjustment (Base \rightarrow w.o. FDA); and 3) heterogeneous teachers (Homogeneous (Ho) Teachers \rightarrow CausalD).

Model	Amazon			Alipay		
	AUC	NDCG	Recall	AUC	NDCG	Recall
DIN (Base)	0.8873	0.5384	0.6877	0.7691	0.2076	0.3518
Ho Teachers	0.8869	0.5429	0.6883	0.7723	0.2289	0.3953
w.o. FDA	0.8952	0.5527	0.7008	0.7727	0.2238	0.4168
CausalD	0.9027	0.5751	0.7216	0.7777	0.2779	0.4457

TABLE 6: Ablation study with DeepFM as the base model.

Model	Amazon			Alipay		
	AUC	NDCG	Recall	AUC	NDCG	Recall
DeepFM (Base)	0.8941	0.5416	0.6877	0.8018	0.2501	0.4177
Ho Teachers	0.8550	0.4371	0.5817	0.7797	0.2669	0.4265
w.o. FDA	0.8860	0.5528	0.6953	0.7811	0.2817	0.4444
CausalD	0.8948	0.5638	0.7016	0.8031	0.3053	0.4657

TABLE 7: Recommendation performance of CausalD as changing the number of heterogeneous teachers.

#Teachers	Amazon			Alipay		
	AUC	NDCG	Recall	AUC	NDCG	Recall
2	0.8910	0.5304	0.6806	0.7866	0.2682	0.4171
4	0.8869	0.5429	0.6883	0.7775	0.2812	0.4351
8	0.8965	0.5652	0.7114	0.7834	0.2902	0.4498
16	0.8931	0.5615	0.7062	0.7723	0.2832	0.4514

back-door adjustment loss and the front-door adjustment loss improve the performance. Specifically, we construct **w.o. FDA** which denotes CausalD trained without front-door adjustment loss. By removing the back-door adjustment loss from **w.o. FDA**, we obtain the **Base** (DIN) model. Model **Ho Teachers** means replacing the heterogeneous teachers in CausalD with homogeneous teachers, which are pre-trained on randomly split data.

According to the results shown in Table 5, we observe that removing any of these three components (heterogeneous teachers, BDA loss, FDA loss) will incur a perfor-

mance degradation. Notably, removing FDA loss ($\text{CausalD} \rightarrow w.o \text{ FDA}$) leads to NDCCG - 19.5% relatively on the AliPay dataset. We attribute the result to the effective capability of front-door adjustment in reducing confounding effects. The performance drop caused by removing BDA loss ($w.o \text{ FDA} \rightarrow \text{Base}$) demonstrates that the back-door adjustment loss is helpful for causal intervention. Not surprisingly, CausalD trained with homogeneous teachers achieves inferior results, which show sampling heterogeneous mediator values given the input is essential for FDA. These results jointly demonstrate the rationality of our framework design. We have mostly similar observations with DeeFM as the base model, according to the results listed in Table 6.

Effect of the Teacher Number. To investigate how the number of teachers affects the performance, we consider some variants of CausalD trained with $\{2, 4, 8, 16\}$ teachers, respectively. Table 7 summarizes the results. By jointly analyzing Table 7 and Table 3, we have the following findings:

- Increasing the number of teachers mostly leads to performance gain. A larger number of teachers means that CausalD conducts more comprehensive sampling on the counterfactual space and thus estimates the causal effect in Equation (5) more accurately.
- When further increasing the number of teachers from 8 to 16, we observe little performance gain in many cases. We attribute this phenomenon to that teachers themselves might become less effective with smaller training datasets. In larger dataset Alipay, CausalD achieves the best Recall.
- When varying the number of teachers, CausalD consistently outperforms the base model DIN. These results demonstrate that the front-door adjustment and the causal distillation can facilitate recommendation.

Effect of the Loss Coefficients. CausalD introduces two losses, *i.e.*, the back-door adjustment loss, and the front-door adjustment loss. To investigate how the loss coefficients (λ_{BDA} and λ_{FDA} in Equation (22)) affect the performance, we vary λ_{BDA} and λ_{FDA} among $\{0.01, 0.1, 1, 10\}$, and test the performance. According to Figure 5, we can find that 1) the recommendation performance is mostly insensitive to these hyper-parameters. Directly leaving them as $\lambda_{BDA} = \lambda_{FDA} = 1$ will achieve substantial improvements over the base model; 2) increasing λ_{FDA} consistently leads to performance gains, which again shows the merits of front-door adjustment for recommender systems; and 3) increasing λ_{BDA} are mostly helpful while relatively large λ_{BDA} might be inferior. This is reasonable since \mathcal{L}_{BDA} is an auxiliary loss for \mathcal{L}_{FDA} . Larger \mathcal{L}_{BDA} might incur less model optimization on \mathcal{L}_{FDA} .

6 CONCLUSION AND FUTURE WORK

In this work, we study how to alleviate the performance heterogeneity problem in recommender systems. Though the *natural source* of this problem, *i.e.*, imbalance of training data distribution over users, is inevitable, we find that model training enlarges the heterogeneity, *i.e.*, a *model source*. By analyzing the recommendation data generation process with the causal graph, we find that there are unobserved confounders that mainly mislead the estimation $P(Y | X)$ of tail users. Different from existing works that require

unbiased data or assume unconfoundedness, we handle unobserved confounders using FDA. We propose the causal multi-teacher distillation framework (CausalD) as an efficient approximation of FDA by distilling causal effects estimated by heterogeneous teachers into a recommendation model. We conduct experiments on three public datasets, validating the effectiveness of CausalD on improving recommendation and alleviating performance heterogeneity.

To the best of our knowledge, this work takes the initiative to incorporate FDA into recommendation models. FDA has the enticing merit to deal with observed confounders that cause the spurious correlation of historical behaviors and next behaviors. Bias is ubiquitous and various in kind, and we believe this new paradigm is general for alleviating model training biases in a unified framework. There are many research directions remaining for exploration. Apart from performance heterogeneity, there are more issues caused by biased model training, such as filter bubble. We are interested in whether CausalD can alleviate these issues in the future. Another future direction relates to performing FDA on causally discovered causal graphs. Lastly, we plan to extend CausalD to state-of-the-art recommendation models that leverage various content features and side information.

ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China (62006207, U20A20387, 62037001), Key R & D Projects of the Ministry of Science and Technology (2020YFC0832500), Young Elite Scientists Sponsorship Program by CAST (2021QNRC001), Zhejiang Province Natural Science Foundation (LR19F020006, LQ21F020020), Project by Shanghai AI Laboratory (P22KS00111), Program of Zhejiang Province Science and Technology (2022C01044), the StarryNight Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-0010).

REFERENCES

- [1] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, 2018.
- [2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web, WWW*, 2017.
- [3] H. Yin, Q. Wang, K. Zheng, Z. Li, and X. Zhou, "Overcoming data sparsity in group recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3447–3460, 2022.
- [4] C. Gao, N. Li, T. Lin, D. Lin, J. Zhang, Y. Li, and D. Jin, "Social recommendation with characterized regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2921–2933, 2022.
- [5] M. Unger and A. Tuzhilin, "Hierarchical latent context representation for context-aware recommendations," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3322–3334, 2022.
- [6] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015.
- [7] H. Steck, "Calibrated recommendations," in *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys*, 2018.
- [8] Y. Wang, X. Wang, A. Beutel, F. Prost, J. Chen, and E. H. Chi, "Understanding and improving fairness-accuracy trade-offs in multi-task learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, 2021.

- [9] Y. Li, H. Chen, Z. Fu, Y. Ge, and Y. Zhang, "User-oriented fairness in recommendation." in *WWW '21: The Web Conference*, 2021.
- [10] J. Pearl, M. Glymour, and N. P. Jewell, *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.
- [11] X. Zheng, M. Wang, R. Xu, J. Li, and Y. Wang, "Modeling dynamic missingness of implicit feedback for sequential recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 405–418, 2022.
- [12] Q. Cui, S. Wu, Q. Liu, W. Zhong, and L. Wang, "MV-RNN: A multi-view recurrent neural network for sequential recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 2, pp. 317–331, 2020.
- [13] W. Fan, Y. Ma, Q. Li, J. Wang, G. Cai, J. Tang, and D. Yin, "A graph neural network framework for social recommendations," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2033–2047, 2022.
- [14] Y. Wang, D. Liang, L. Charlin, and D. M. Blei, "The deconfounded recommender: A causal inference approach to recommendation." *CoRR*, 2018.
- [15] A. Agarwal, K. Takatsu, I. Zaitsev, and T. Joachims, "A general framework for counterfactual learning-to-rank." in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, 2019.
- [16] W. Wang, F. Feng, X. He, X. Wang, and T.-S. Chua, "Deconfounded recommendation for alleviating bias amplification," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, 2021.
- [17] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, "Causal intervention for leveraging popularity bias in recommendation." in *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [18] W. Wang, F. Feng, X. He, H. Zhang, and T.-S. Chua, "Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue." in *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [19] D. Liu, P. Cheng, Z. Dong, X. He, W. Pan, and Z. Ming, "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, SIGIR*, 2020.
- [20] D. B. Rubin, "Bayesian inference for causal effects: The role of randomization," *The Annals of statistics*, 1978.
- [21] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network." *CoRR*, vol. abs/1503.02531. [Online]. Available: <http://arxiv.org/abs/1503.02531>
- [22] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback." in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI*, 2009.
- [23] Y. Bengio and J.-S. Senecal, "Adaptive importance sampling to accelerate training of a neural probabilistic language model." *IEEE Trans. Neural Networks*, 2008.
- [24] J. Chen, Y. Feng, M. Ester, S. Zhou, C. Chen, and C. Wang, "Modeling users' exposure with social knowledge influence and consumption influence for recommendation." in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM*, 2018.
- [25] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, "Modeling user exposure in recommendation." in *Proceedings of the 25th International Conference on World Wide Web, WWW*, 2016.
- [26] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, "Recommendations as treatments: Debiasing learning and evaluation." in *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 2016.
- [27] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling user interest and conformity for recommendation with causal embedding." in *WWW '21: The Web Conference*, 2021.
- [28] A. Agarwal, I. Zaitsev, X. Wang, C. Li, M. Najork, and T. Joachims, "Estimating position bias without intrusive interventions." in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM*, 2019.
- [29] X. Wang, R. Zhang, Y. Sun, and J. Qi, "Doubly robust joint learning for recommendation on data missing not at random." in *Proceedings of the 36th International Conference on Machine Learning, ICML*, 2019.
- [30] Z. Zhao, J. Chen, S. Zhou, X. He, X. Cao, F. Zhang, and W. Wu, "Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation." *CoRR*, vol. abs/2109.07946, 2021.
- [31] Y. Wang, D. Liang, L. Charlin, and D. M. Blei, "Causal inference for recommender systems." in *RecSys 2020: Fourteenth ACM Conference on Recommender Systems*, 2020.
- [32] J. Jin, Y. Fang, W. Zhang, K. Ren, G. Zhou, J. Xu, Y. Yu, J. Wang, X. Zhu, and K. Gai, "A deep recurrent survival model for unbiased ranking." in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR*, 2020.
- [33] K. Nagatani and M. Sato, "Accurate and diverse recommendation based on users' tendencies toward temporal item popularity." in *Proceedings of the 1st Workshop on Temporal Reasoning in Recommender Systems co-located with 11th International Conference on Recommender Systems*, 2017.
- [34] S. Bonner and F. Vasile, "Causal embeddings for recommendation." in *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys*, 2018.
- [35] T. Wei, F. Feng, J. Chen, Z. Wu, J. Yi, and X. He, "Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system," in *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18*, 2021. ACM, 2021, pp. 1791–1800.
- [36] D. Liu, P. Cheng, H. Zhu, Z. Dong, X. He, W. Pan, and Z. Ming, "Mitigating confounding bias in recommendation via information bottleneck." in *RecSys '21: Fifteenth ACM Conference on Recommender Systems*, 2021.
- [37] Y. He, Z. Wang, P. Cui, H. Zou, Y. Zhang, Q. Cui, and Y. Jiang, "Causpref: Causal preference learning for out-of-distribution recommendation." in *WWW '22: The ACM Web Conference 2022*. ACM, 2022, pp. 410–421.
- [38] Y. Wang, H. Guo, B. Chen, W. Liu, Z. Liu, Q. Zhang, Z. He, H. Zheng, W. Yao, M. Zhang, Z. Dong, and R. Tang, "Causalint: Causal inspired intervention for multi-scenario recommendation," in *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18*, 2022. ACM, 2022, pp. 4090–4099.
- [39] L. Yang, Y. Cui, Y. Xuan, C. Wang, S. J. Belongie, and D. Estrin, "Unbiased offline recommender evaluation for missing-not-at-random implicit feedback." in *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys*, 2018.
- [40] "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer." in *5th International Conference on Learning Representations, ICLR*, 2017.
- [41] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation." in *2019 IEEE/CVF International Conference on Computer Vision, ICCV*, 2019.
- [42] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation." in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.
- [43] B. Peng, X. Jin, D. Li, S. Zhou, Y. Wu, J. Liu, Z. Zhang, and Y. Liu, "Correlation congruence for knowledge distillation." in *2019 IEEE/CVF International Conference on Computer Vision, ICCV*, 2019.
- [44] Z. Shen, Z. He, and X. Xue, "Meal: Multi-model ensemble via adversarial learning." in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, 2019.
- [45] X. Tan, Y. Ren, D. He, T. Qin, Z. Zhao, and T.-Y. Liu, "Multilingual neural machine translation with knowledge distillation." in *7th International Conference on Learning Representations, ICLR*, 2019.
- [46] J. Zhu, J. Liu, W. Li, J. Lai, X. He, L. Chen, and Z. Zheng, "Ensembled ctr prediction via knowledge distillation." in *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*, 2020.
- [47] S. Wang, K. Zhang, L. Wu, H. Ma, R. Hong, and M. Wang, "Privileged graph distillation for cold start recommendation." in *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [48] J.-w. Lee, M. Choi, J. Lee, and H. Shim, "Collaborative distillation for top-n recommendation." in *2019 IEEE International Conference on Data Mining, ICDM*, 2019.
- [49] S. Kang, J. Hwang, W. Kweon, and H. Yu, "Topology distillation for recommender system." in *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021.
- [50] Q. Wang, H. Yin, T. Chen, Z. Huang, H. Wang, Y. Zhao, and N. Q. V. Hung, "Next point-of-interest recommendation on resource-constrained mobile devices." in *WWW '20: The Web Conference*, 2020.

- [51] X. Chen, Y. Zhang, H. Xu, Z. Qin, and H. Zha, "Adversarial distillation for efficient recommendation with external knowledge." *ACM Trans. Inf. Syst.*, 2019.
- [52] J. Yao, F. Wang, K. Jia, B. Han, J. Zhou, and H. Yang, "Device-cloud collaborative learning for recommendation." in *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, 2021.
- [53] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering." in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, 2019.
- [54] D. Liang, L. Charlin, and D. M. Blei, "Causal inference for recommendation," in *Causation: Foundation to Application, Workshop at UAI, AUAI*, 2016.
- [55] M. Lukasik, S. Bhojanapalli, A. K. Menon, and S. Kumar, "Teacher's pet: understanding and mitigating biases in distillation." *CoRR*, 2021.
- [56] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for ctr prediction." in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, 2017.
- [57] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *J. Mach. Learn. Res.*, 2011.
- [58] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans." in *Advances in Neural Information Processing Systems*, 2016.



Fuli Feng is a professor at the University of Science and Technology of China (USTC). He received Ph.D. in Computer Science from NUS in 2019. His research interests include information retrieval, data mining, and multi-media processing. He has over 30 publications appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE and TOIS. His work on Bayesian Personalized Ranking has received the Best Poster Award of WWW 2018. Moreover, he has been served as the PC member for several top conferences including SIGIR, WWW, WSDM, NeurIPS, AAAI, ACL, MM, and invited reviewer for prestigious journals such as TOIS, TKDE, TNNLS, TPAMI, and TMM.



Kun Kuang received his Ph.D. degree from Tsinghua University in 2019. He is now an Associate Professor in the College of Computer Science and Technology, Zhejiang University. He was a visiting scholar with Prof. Susan Athey's Group at Stanford University. His main research interests include Causal Inference, Artificial Intelligence, and Causally Regularized Machine Learning. He has published over 40 papers in major international journals and conferences, including SIGKDD, ICML, ACM MM, AAAI, IJCAI, TKDE, TKDD, Engineering, and ICDM, etc.



Shengyu Zhang is currently a Ph.D. student at Zhejiang University, Hangzhou, China. He got the China Scholarship Council Fellowship in 2020 and visited National University of Singapore, Singapore, as a visiting research scholar from 2021 to 2022 (expected). His current research interests are in information retrieval, and causally regularized machine learning. Until now, he has published more than 20 papers in major international journals and conferences such as SIGKDD, SIGIR, ACM MM, WWW, CVPR, AAAI, and TKDE, etc. He also serves as Reviewer of several high-level international journals such as TKDE, TCYB, TNNLS, etc.



Zhou Zhao (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from The Hong Kong University of Science and Technology, in 2010 and 2015, respectively. He is currently an Associate Professor with the College of Computer Science, Zhejiang University. His research interests include machine learning and data mining.



Ziqi Jiang is currently an undergraduate student at the College of Computer Science, Zhejiang University, advised by Prof. Fei Wu. His research interests lie at the intersection of multimedia, AI security, and causal inference, with the goal to build socially efficient and robust machine learning systems. He has published papers in ACM MM.



Shuo Li is an undergraduate at School of Computing, National University of Singapore (NUS), and works as an research intern at Lazada data science team. His research interests include multi-modal learning and natural language processing.



Jiangchao Yao is an Assistant Professor at Shanghai Jiao Tong University. He got his dual Ph.D. degree under the supervision of Ya Zhang in Shanghai Jiao Tong University and Ivor W. Tsang in University of Technology Sydney in 2019. His research interests lie on Trustworthy Machine Learning and Reasoning, making machine learning robust, interpretable and scalable. He has published over 20 papers in major international journals and conferences such as ICML, ICLR, NeurIPS, KDD, TPAMI, TIP, TKDE, etc..



Hongxia Yang received the PhD degree in statistics from Duke University, in 2010. She is working as a senior staff data scientist and director with the Alibaba Group. She has published more than 60 papers and held nine filed/to be filed US patents and is serving as the associate editor of Applied Stochastic Models in Business and Industry.



Tat-Seng Chua is the KITHCT Chair Professor at the School of Computing, National University of Singapore. He was the Acting and Founding Dean of the School during 1998-2000. Dr Chua's main research interest is in multimedia information retrieval and social media analytics. In particular, his research focuses on the extraction, retrieval and question-answering (QA) of text and rich media arising from the Web and multiple social networks. He is the co-Director of NExT, a joint Center between NUS and Tsinghua

University to develop technologies for live social media search. Dr Chua is the 2015 winner of the prestigious ACM SIGMM award for Outstanding Technical Contributions to Multimedia Computing, Communications and Applications. He is the Chair of steering committee of ACM International Conference on Multimedia Retrieval (ICMR) and Multimedia Modeling (MMM) conference series. Dr Chua is also the General Co-Chair of ACM Multimedia 2005, ACM CIVR (now ACM ICMR) 2005, ACM SIGIR 2008, and ACM Web Science 2015. He serves in the editorial boards of four international journals. Dr. Chua is the co-Founder of two technology startup companies in Singapore. He holds a PhD from the University of Leeds, UK.



Fei Wu (Senior Member, IEEE) received the Ph.D. degree from Zhejiang University, Hangzhou, China. He was a Visiting Scholar with the Prof. B. Yu's Group, University of California at Berkeley, Berkeley, from 2009 to 2010. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. His current research interests include multimedia retrieval, sparse representation, and machine learning.