Counterfactual Video Recommendation for Duration Debiasing

Shisong Tang*
Tsinghua University
Beijing, China
tangss21@mails.tsinghua.edu.cn

Qing Li†
Peng Cheng Laboratory
Shenzhen, China
liq@pcl.ac.cn

Dingmin Wang
University of Oxford
Oxford, United Kingdom
dingmin.wang@cs.ox.ac.uk

Ci Gao
Jilin University
Changchun, China
gaoci21@mails.jlu.edu.cn

Wentao Xiao
Tsinghua University
Beijing, China
xwt20@mails.tsinghua.edu.cn

Dan Zhao
Peng Cheng Laboratory
Shenzhen, China
zhaod01@pcl.ac.cn

Yong Jiang
Tsinghua University
Beijing, China
jiangy@sz.tsinghua.edu.cn

Qian Ma
ByteDance Inc.
Beijing, China
maqian.zero@bytedance.com

Aoyang Zhang
ByteDance Inc.
Beijing, China
zhangaoyang@bytedance.com

ABSTRACT

Duration bias widely exists in video recommendations, where models tend to recommend short videos for the higher ratio of *finish playing* and thus possibly fail to capture users' true interests. In this paper, we eliminate the duration bias from both data and model. First, based on the extensive data analysis, we observe that *play completion rate* of videos with the same duration presents a bimodal distribution. Hence, we propose to perform threshold division to construct binary labels as training labels for alleviating the drawback of *finish playing* labels overly biased towards short videos. Algorithmically, we resort to causal inference, which enables us to inspect causal relationships of video recommendations with a causal graph. We identify that duration has two kinds of effect on prediction: direct and indirect. Duration bias lies in the direct effect, while the indirect effect benefits prediction. To this end, we design a model-agnostic Counterfactual Video Recommendation for Duration Debiasing (CVRDD) framework, which incorporates multi-task learning to estimate different causal effect during training. In the inference phase, we perform counterfactual inference to remove the direct effect of duration for unbiased prediction. We conduct experiments on two industrial datasets, and in addition to achieving highly promising results on traditional top-k recommendation metrics, CVRDD also improves the user watch time.

CCS CONCEPTS

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

KEYWORDS

Recommender System, Counterfactual Inference, Debias

*Work done at ByteDance Inc.
†Corresponding author.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599797

ACM Reference Format:
Shisong Tang, Qing Li, Dingmin Wang, Ci Gao, Wentao Xiao, Dan Zhao, Yong Jiang, Qian Ma, and Aoyang Zhang. 2023. Counterfactual Video Recommendation for Duration Debiasing. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599797

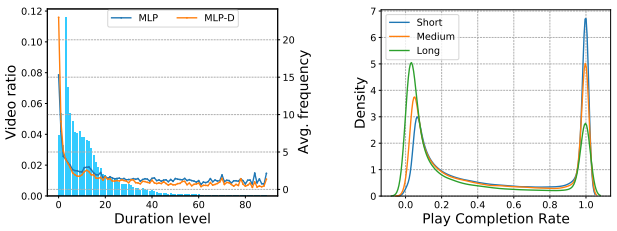
1 INTRODUCTION

With the prevalence of Web 2.0 and mobile devices, an increasing number of users are joining online video platforms such as Youtube[1], TikTok[2], and Douyin[3] for sharing and viewing. In scenarios of video recommendations, previous studies focus on fitting historical interactions to learn the matching score between user preference and item feature [1, 4], where *finish playing* [7, 12] is widely used as the training label. However, the presence of the duration feature may make the prediction of such data-driven recommendation models biased. For example, short videos are more likely to finish, even if users are not interested in the video content. The spurious correlation between video duration and interaction makes predictions deviate from users' real preference [12, 41].

Figure 1(a) shows the duration bias on the ByteDance data, where we train a MLP [4] and count the frequency of videos in the top-20 recommendation lists of all users. Blindly fitting data makes short videos over-recommended. Worse, the model is more susceptible to attacks, such as video creators who may intentionally post short videos to obtain more exposure. Therefore, it is crucial to remove the shortcut between duration and prediction to build personalized video recommendation systems.

A straightforward solution is to use post-feedback (e.g., comments and likes) instead of *finish playing* as training labels [39]. However, in the real scenarios, we observe that videos with post-feedback are extremely sparse, so only collecting these videos for training means that we will lose a large proportion of positive samples [35]. Existing solutions are mainly from a causal perspective to debias. Inverse Propensity Weighting (IPW) [19, 29–31] adjusts

[1]https://www.youtube.com/
[2]https://www.tiktok.com/
[3]https://www.douyin.com/



(a) (b)

Figure 1: (a) An illustration of duration bias in recommender system. The blue histogram indicates the ratio of videos in different duration levels. The curve represents the average recommendation frequency of models. (b) PCR distribution of three different duration levels.

the data distribution by re-weighting the training samples to make them unbiased. However, inaccuracy of the estimated propensity scores may lead to high variance [1, 40, 42]. Some recent works applied causal intervention with backdoor adjustment [27] over the causal graph to mitigate the bias [12, 34, 41, 42]. Different from these previous works, we design a novel framework with counterfactual inference to perform duration debiasing.

First of all, based on extensive data analysis, we found that users' *play completion rate* (PCR) show a bimodal distribution under the same video duration. Figure 1(b) shows the distribution of PCR under three different duration levels on the ByteDance data. This statistical phenomenon is also consistent with the behavior of most people in watching TikTok, who directly cross out the videos they don't like and might watch multiple times for the content they prefer. Therefore, based on such natural properties of PCR, we can easily perform threshold division to obtain binary labels to alleviate the over-bias of *finish playing* as a training label for short videos. Specifically, we obtain the threshold using Otsu's algorithm [24], which is a widely used image segmentation method when the image gray histogram shows a bimodal distribution.

Although using binary PCR as training labels can help alleviate the issue of duration bias, this kind of data-level optimization is still incapable of eliminating the shortcut between video duration and prediction. To this end, we resort to causal inference [27], in which we formally define the video recommendation process as a causal graph. From the graph, we find two effects of video duration on model prediction: *direct* and *indirect*. The *direct* effect refers to a shortcut from duration to prediction, showing spurious correlations in duration (i.e., undesirable effects). For *indirect* effect, it extracts reliable information by considering user-item matching scores. For example, when two videos relatively match a user's interest, the shorter video may be more likely to finish playing.

To eliminate the direct effect of duration on prediction and retain the indirect effect it brings, we propose a model-agnostic counterfactual video recommendation framework (CVRDD). Specifically, CVRDD incorporates duration into the matching score modeling to retain the indirect effect. Meanwhile, we design a residual module to separately model the shortcut from duration to prediction during training. CVRDD estimates the direct effect by counterfactual inference in the inference phase, i.e., imagining what the prediction would be if the model only saw the effect of duration. CVRDD achieves duration debiasing by retaining the indirect effect of duration and removing the direct effect during the inference phase. In the experiments, we use binary PCR for training and post-feedback labels for evaluation that can reflect the real user preference. The results show that CVRDD not only achieves good results on Top-k recommendation metrics, but also improves on user watch time. Our contributions are summarized as follows:

- We formulate duration bias in video recommendation as causal effects and propose a model-agnostic counterfactual video recommendation for duration debiasing framework.
- To alleviate over-bias of *finish playing* labels on short videos, we propose to perform threshold division to construct binary training labels based on the *play completion rate*.
- Extensive experiments on two datasets collected from two large-scale video-sharing platforms demonstrate the effectiveness of our proposed duration debias scheme.

2 RELATED WORK

2.1 Bias in Recommendation

Recently, several different types of biases have been studied in the recommendation system [1]. The first type of biases originates from experimental data. In other words, the distribution of the training data is different from the ideal test data distribution. Selection bias happens as users are free to choose which items to rate, so that the observed ratings are not a representative sample of all ratings [13, 22, 33]. Exposure bias occurs when users are exposed to a part of specific items, so the unobserved interactions do not always represent negative preference [2, 21, 25]. Conformity bias results from group psychology in that users follow others in a group even if the choice goes against their propensity [17, 18, 36]. Position bias happens as users prefer to interact with those items in a higher position of the list regardless of the items' actual relevance [3, 14, 15]. The second type of biases comes from the model. Inductive bias denotes the assumptions made by the model to better learn the target function and to generalize beyond training data [5, 26]. The last type of bias in recommendation includes popularity bias and unfairness. Popularity bias happens when a small fraction of the whole items gain more popularity recurrently due to the long tail phenomenon [38, 42]. Unfairness means the predilection of the recommendation system discriminates against certain users or groups systematically [6, 20].

2.2 Causal Recommendation

Recently, causal inference has become a hot topic in the recommendation system [1]. The most popular methods can be divided into three types. **Inverse Propensity Weighting** (IPW) adjusts the training distribution by re-weighting training samples with propensity scores [19, 29–31]. However, inaccuracy of the estimated propensity scores may lead to high variance [1, 40, 42]. **Causal Intervention**, where researchers intervene the causal relationship that causes the bias and add adjustments to eliminate this harmful effect for more accurate estimation [12, 34, 37, 41, 42]. Among them, PD [42] is proposed to solve the popularity bias problem and DecRS [34] is used in the bias amplification problem. [12, 41] adopts the backdoor adjustment to solve duration bias. However, because the sample space is too large, their approximation of scores over the

intervention terms is subject to large variance and lacks stability. **Counterfactual Inference** adjusts prediction score by reducing the bad effect of bias [23]. [35] is to solve the clickbait issue, and [38] is to eliminate the popularity bias.

3 PRELIMINARIES

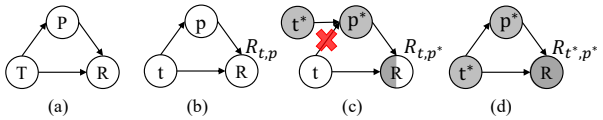


Figure 2: The causal graph of placebo effect in drug treatment. T, P, and R stand for treatment, placebo, and recovery, respectively. White nodes are at the value $P = p$ while gray nodes mean the variables are at reference status (e.g., $P = p^*$).

In this section, we recapitulate some key concepts and theories about causal inference [27, 28, 40].

Causal Graph. A causal graph is a Directed Acyclic Graph consisting of nodes and edges, where nodes represent random variables and edges represent causal relationships between two connected variables. Figure 2(a) illustrates an example of causal graph, where random variables T, P, and R stand for treatment, placebo, and recovery, respectively. The lowercase letters denote the specific values of random variables. $T \rightarrow R$ means that treatment T has a direct effect on recovery R. $T \rightarrow P \rightarrow R$ means that placebo P acts as a mediator [28] between the indirect effects of treatment T and recovery R.

Counterfactual Inference. Counterfactual inference is a thinking activity that denies facts that have already occurred and re-represents them in order to construct a possibility hypothesis. In the example shown in Figure 2(b), it emphasizes *"what would happen if the treatment was the exact opposite of reality"* and can be used to compare the results of interventions in complex systems. Formally, the structural equation of the causal graph is defined as follows:

$$R_{t,p} = R(T = t, P = p), \quad p = P_t = P(T = t), \tag{1}$$

where $R(\cdot)$ and $P(\cdot)$ are the structural equations of R and P, respectively, which can be learned from the observed data. P_t indicates what the placebo of someone would be if he/she has the treatments t. $R_{t,p}$ denotes what the recovery outcome of someone would be if he/she has the placebo p and treatment t.

Figure 2(c) shows the causal graph of the factual and counterfactual worlds, with the gray nodes representing the random variables in the reference state (e.g. $P = p^*$). When T is in the reference state, P is set to $p^* = P(T = t^*)$ and R is set to $R_{t,p^*} = R(T = t, P = p^*)$. Due to the presence of p^* and t, the color of node R is set to half gray, which is called the counterfactual scenario because it doesn't really happen in the factual world. It is only imagined in order to study what the final result would be if T was simultaneously set to different values of t and t^*.

Causal Effect. Formally, causal effect can be defined as the difference between the outcome of the counterfactual world and the outcome of the real-world observations for the same individual [27]. The total effect (TE) of treatment $T = t$ on R compares these two situations $T = t$ and $T = t^*$, which is denoted as:

$$TE = R_{t,p} - R_{t^*,p^*}. \tag{2}$$

Total effect can be regarded as the sum of natural direct effect (NDE) and total indirect effect (TIE). NDE represents the effect of T on R when the mediator P is blocked. It expresses the increase in the recovery R with T changing from t^* to t under the pure environment $P(T = t^*)$:

$$NDE = R_{t,p^*} - R_{t^*,p^*}. \tag{3}$$

Similarly, TIE reflects the effect of T on R through the mediator P. TIE is the difference between TE and NDE, denoted as:

$$TIE = TE - NDE = R_{t,p} - R_{t,p^*}. \tag{4}$$

4 DEBIAS FRAMEWORK

We first give the definition of the task in Section 4.1, then Section 4.2 presents the scheme for threshold division for PCR, Section 4.3 looks at the duration bias from a causal perspective and introduces the CVRDD framework, and Section 4.4 presents the details of instantiation, training and inference for CVRDD.

4.1 Task Fomulation

Let $\mathcal{U} = \{u_1, ..., u_N\}$, $\mathcal{V} = \{v_1, ..., v_M\}$ and $D = \{d_1, ..., d_L\}$ denote the set of users, items and duration levels, respectively, where N is the number of users, M is the number of items, and L is the number of duration levels. The user-item historical interactions are represented by $\mathcal{D} = \{(u, v, d, y) | u \in \mathcal{U}, v \in \mathcal{V}, d \in D\}$, where $y \in \{0, 1\}$ denotes the binary label (e.g., finish playing or like). The target of the traditional video recommendation training is to learn the scoring function $f(u, i|\Theta)$ from $\mathcal{D}$, which is capable of predicting the preference of user u on item i, where Θ is the parameters of f.

4.2 Binary PCR Label

A user's play is defined as *finish playing* if its duration is greater than or equal to the video duration. Obviously, *finish playing* has serious bias towards short videos, and the model trained with *finish playing* as its ground-truth label will exacerbate the preference for short videos. To mitigate the bias of *finish playing* label at the data level, we introduce **P**lay **C**ompletion **R**ate (PCR) defined as:

$$\text{PCR} = \min\left(1.0, \frac{Play\ Duration}{Video\ Duration}\right). \tag{5}$$

Figure 1(b) shows the distribution of PCR for three types of video durations on ByteDance data (video durations have been discretized). As can be seen, the distribution of PCR of each video duration type presents a bimodal distribution. Moreover, as the video duration increases, the left peak increases while the right peak decreases. This is also consistent with the perception that short videos are more likely to finish broadcasting and therefore have a higher right peak.

Exploiting the natural property of bimodal distribution, we can easily perform threshold division to obtain binary labels. Inspired by the Otsu algorithm [24] in image segmentation, we design a threshold division scheme for bimodal distribution as follows.

First, we discretize the PCR arrays into S bins, each with probability p_s. The threshold $t \in [1, S]$ divides them into two classes: C_0 and C_1. Class C_0 has a probability of $w_0 = \sum_{i=1}^{t} p_i$, a mean of u_0 and a variance of σ_0^2. Class C_1 has a probability of $w_1 = \sum_{i=t+1}^{S} p_i$, a mean of u_1 and a variance of σ_1^2.

The optimal division should achieve minimum intra-class variance and maximum inter-class variance. Otsu algorithm proves that maximizing the inter-class variance is equivalent to minimizing the intra-class variance. Therefore, we expect to maximize the inter-class variance σ_b^2, which is given as:

$$\sigma_b^2 = w_0(u_0 - u)^2 + w_1(u_1 - u)^2, \tag{6}$$

where u is the overall mean of the two classes, which can be calculated as $u = w_0 u_0 + w_1 u1$. As such, (6) can be rewritten as:

$$\sigma_b^2 = w_0 w_1 (u_0 - u_1)^2. \tag{7}$$

Due to the large bimodal difference of the PCR distribution, we also wish to minimize the mean standard deviation $\overline{\sigma}$ in order to make the intra-classes more compact, where $\overline{\sigma}$ is given as:

$$\overline{\sigma} = w_0 \sigma_0 + w_1 \sigma_1. \tag{8}$$

Finally, incorporating both objectives above, the best division threshold t^* is obtained as:

$$t^* = \max_{t \in \{1,\ldots,S\}} \frac{w_0 w_1 (u_0 - u_1)^2}{w_0 \sigma_0 + w_1 \sigma_1}. \tag{9}$$

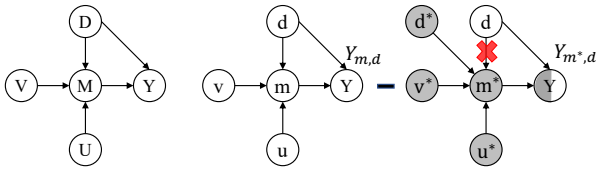
4.3 Counterfactual Video Recommendation.

In this section, we first introduce the duration bias in video recommender systems through cause graphs. Then, we propose CVRDD, a framework for eliminating the duration bias.

4.3.1 ***Casual graph***. As shown in Figure 3(a), we use a causal graph to describe the causal relationships in the video recommendation task, i.e., the relationships among user U, video V, match score M, video duration D, and model prediction Y. The explanation of these causal relationships is as follows:

- U denotes user features, e.g., ID, location.
- V denotes video representation, e.g., ID, category, author.
- D denotes video duration which is a confounding feature.
- M represents the match score, which reflects the user's preference for the video.
- Y represents the prediction of whether the user interacts with the video, such as finish playing and likes.
- $\{U, V\} \rightarrow M \rightarrow Y$ represents the indirect effect of U, V on Y through the mediator variable M. This path is to develop a statistical model for the conditional probability $P(Y|U, V)$.
- $\{U, V, D\} \rightarrow M \rightarrow Y$ targets at modeling $P(Y|U, V, D)$ via extending tranditional recommendation model, which exploits the video duration feature.
- $D \rightarrow Y$ represents a shortcut between video duration and interaction, as video duration provides salient features with spurious correlation for interaction.

From the causal graph, it can be concluded that D affects Y from two aspects: 1) the indirect effect of D on Y via mediator M, which is useful for prediction; and 2) the direct effect of D on Y, which makes the relationship between interaction and match score spurious, thus creating the duration bias. Therefore, to eliminate the duration bias, it is necessary to remove the direct effect of video duration on the interaction, i.e., the path $D \rightarrow Y$.



(a) Causal graph (b) Counterfactual inference

Figure 3: (a) Causal graph for Video Recommendation tasks. (b) Illustration of counterfactual inference.

4.3.2 ***CVRDD framework***. Video recommendation model predicts interaction behavior Y from the match scores of user u and video v, i.e., path $\{U, V, D\} \rightarrow M \rightarrow Y$ in the causal graph, which is modeled as $P(Y|U, V, D)$. A good video recommendation model P should have prediction results that reflect the real preferences of users. According to the previous analysis, model P mixes the direct and indirect effect leading to duration bias. Therefore, we propose CVRDD to remove the direct effect of duration D on the interaction Y. To further analyze the causes of the interaction, we describe the form of the interaction Y based on user u and video v with the duration d as:

$$Y_{m,d} = f_Y(M = m, D = d), \tag{10}$$

where f_Y is a function that combines the match score and the duration score. The match score m is calculated as:

$$m = Y_m = f_M(U = u, V = v, D = d), \tag{11}$$

where f_M is an arbitrary video recommendation model, such as DeepFM [8]. Following the definition of causal effects in Section 3, if a no-treatment $D = d^*$ is applied on both the direct and indirect effects, the total effect (TE) is:

$$TE = Y_{m,d} - Y_{m^*,d^*}. \tag{12}$$

The natural direct effect (NDE) of the duration feature of $D = d$ is:

$$NDE = Y_{m*,d} - Y_{m^*,d^*}, \tag{13}$$

where $Y_{m^*,d}$ answers a counterfactual problem: what the model prediction would be if the VR model only had seen the direct effect of duration. According to Eq. (4) in Section 3, the natural approach is to use the total indirect effect (TIE) for inference, i.e., subtract the natural direct effect (NDE) from the total effect (TE):

$$TIE = TE - NDE = Y_{m,d} - Y_{m^*,d}, \tag{14}$$

which mitigates the bad effect (i.e., NDE) caused by the path $D \rightarrow Y$. As such, we utilize the reliable indirect effect (i.e., TIE) for the unbiased prediction by counterfactual inference.

4.4 Instantiation.

To achieve debiasing, CVRDD first needs to be instantiated, and then we introduce the corresponding multi-task training framework and counterfactual inference scheme as shown in Figure 4.

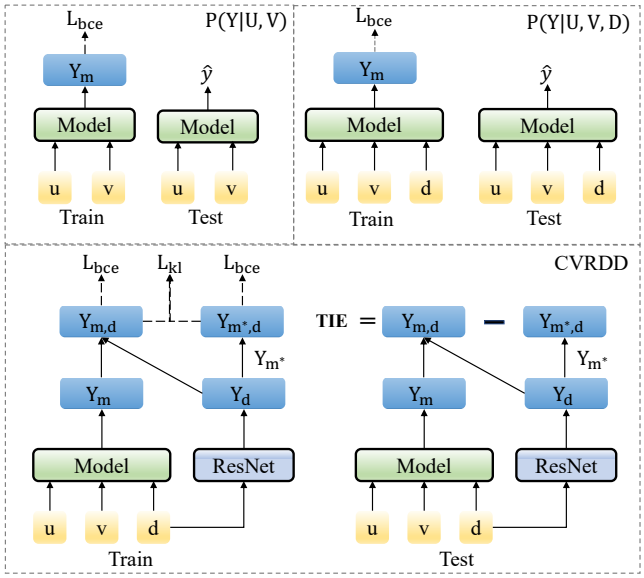


Figure 4: Comparison between our CVRDD and traditional Video Recommendation (VR) methods.

4.4.1 ***Pameterization.*** To instantiate the CVRDD framework, we now need to implement the following functions:

$$Y_m = f_M(U = u, V = v, D = d), \tag{15}$$

where f_M can be any recommendation model. Here, we use a multi-layer perceptron (MLP) to compute the match score Y_m between user u and video v of duration d.

In addition, we adopt a residual network [11] denotes as f_D to calculate the score Y_d of duration $D = d$:

$$Y_d = f_D(D = d). \tag{16}$$

The final perdition score is obtained by aggregating Y_m/Y_{m^*} and Y_d through the aggregation function f_Y:

$$Y_{m,d} = f_Y(M = m, D = d) = \mathcal{F}(Y_m, Y_d), \tag{17}$$

$$Y_{m^*,d} = f_Y(M = m^*, D = d) = \mathcal{F}(Y_{m^*}, Y_d), \tag{18}$$

where $m^* = f_M(U = u^*, V = v^*, D = d^*)$ is the reference state (seeing in Section 3). $\mathcal{F}$ is one of the following three fusion functions:

- Summation [35]

$$\mathcal{F}(Y_m, Y_d) = \log \sigma(Y_m + Y_d), \tag{19}$$

  where σ is the sigmoid function.
- Harmonic [23]

$$\mathcal{F}(Y_m, Y_d) = \log \frac{Y_{HM}}{Y_{HM} + 1}, \tag{20}$$

  where $Y_{HM} = \sigma(Y_m) * \sigma(Y_d)$.
- Multiplication [38]

$$\mathcal{F}(Y_m, Y_d) = Y_m * \sigma(Y_d). \tag{21}$$

Since counterfactual inference takes the maximum prediction as the outcome, whether or not to normalize $Y_{m,d}$ is optional.

4.4.2 ***Multi-task training.*** Based on the CVRDD framework and parameterization scheme, our model requires two prediction scores: $Y_{m,d}$ and $Y_{m^*,d}$. Therefore, we adopt a multi-task learning method to train our model. Since both are binary classification tasks, we take binary cross-entropy (BCE) as the training loss function:

$$L_P = \text{BCE}(Y_{m,d}, y) + \alpha * \text{BCE}(Y_{m^*,d}, y), \tag{22}$$

where α is the weight to balance two tasks, and y is the ground-truth. Figure 4 shows the workflow of traditional video recommendation and CVRDD in the training and test phases.

An unsolved question is how to calculate $Y_{m^*,d} = \mathcal{F}(Y_{m^*}, Y_d)$, where Y_{m^*} denotes model f_M without (u,v) as inputs in the counterfactual world. A straightforward solution is to input u and v as zero-vectors into the model f_M to obtain Y_{m^*}. Instead of that, many previous works [35, 38] choose to ignore Y_{m^*} and only model Y_d as a substitute for $Y_{m^*,d}$. However, such a simplified approach leads to certain inaccuracies. In the real world, humans always tend to guess with a certain probability when it is undetermined with many treatments. Therefore, we introduce a learnable parameter a as Y_{m^*} used to control the sharpness of the distribution of $Y_{m^*,d}$,

$$a = Y_{m^*} = f_M(U = u^*, V = v^*). \tag{23}$$

Since learning is uncontrollable and an unreasonable a would lead to the result that TIE in Eq.(4) is dominated by TE or NDE. Thus, we introduce Kullback-Leibler Divergence to estimate a:

$$L_{kl} = KL(p(y|m, d)||p(y|m*, d)), \tag{24}$$

where $p(y|m, d) = \sigma(Y_{m,d})$ and $p(y|m^*, d) = \sigma(Y_{m^*,d})$. Accordingly, our final training loss is:

$$\mathcal{L} = \sum_{(u,v,d,y)\in\mathcal{D}} L_P + \beta * L_{kl}, \tag{25}$$

where β is the hyper-parameter to balance the KL loss.

4.4.3 ***Inference.*** To eliminate the direct causal effect of path $D \rightarrow Y$, we need to subtract the prediction score $Y_{m,d}$ of model f_M from the prediction score $Y_{m^*,d}$ of model f_D to obtain the final unbiased prediction scores:

$$TIE = Y_{m,d} - Y_{m^*,d} = \mathcal{F}(Y_m, Y_d) - \mathcal{F}(a, Y_d), \tag{26}$$

In this way, we can eliminate the bad effect of duration bias during the stage of inference.

5 EXPERIMENT

5.1 Dataset

1) Wechat: This dataset was adopted in WeChat Big Data Challenge[4], which records the behavior of users on short videos in two weeks. We discretize duration into $L = 6$ levels. The user id, device id, video id, author id, and multimodal content feature vectors are used as other feature inputs.

2) ByteDance[5]: We collect the interaction data of feed streams from the server logs of ByteDance's video platform, which contains 200,000 highly active users for a total of 30 days from April 1, 2022 to April 30, 2022. We discretize duration into $L = 90$ levels. The pretrained ID embeddings and side information are used as other feature inputs for all methods.

[4]https://algo.weixin.qq.com/2021/problem-description
[5]https://www.bytedance.com/en/products

To split the dataset, we sort each user's data chronologically, and divide the training set, validation set, and test set by 6:2:2 per user. We consider two training labels: **Finish Playing (FP)** and **binary PCR**. To evaluate the effectiveness of different debiasing schemes, we follow previous works [12, 35] to construct unbiased test data w.r.t. Duration using post-feedback (comments and likes) as **Test Labels (TL)** which can reflect true user interests. In summary, each interaction has two different training labels and one test label, all of which are binary. Table 1 shows the statistics of two datasets.

Table 1: Statistical Information. X^+/X^- denotes the ratio of positive and negative samples of label X.

Data	#user	#video	#interaction	FP^+/FP^-	PCR^+/PCR^-	TL^+/TL^-
WeChat	20,000	96,428	7,210,290	0.83	1.15	0.05
ByteDance	200,000	9,633,578	183,800,612	0.39	0.94	0.01

5.2 Experimental Setup

5.2.1 ***Metrics.*** To evaluate the performance of different methods, we adopt three widely-used Top-K recommendation metrics: Recall@K, MAP@K, and NDCG@K. We report the results for $K = 3$ and $K = 5$ on WeChat and $K = 10$ and $K = 20$ on ByteDance. In addition, since user watch time is of great importance for video providers, we design a metric: **Average User Time Coverage** (AUTC@K) as follows:

$$AUTC@K = \frac{1}{|U|}\sum_{i=1}^{|U|}\frac{\sum_{j=1}^{K} R_{ij}^{\#time}}{\sum_{k=1}^{M} T_{ik}^{\#time}},$$

where $|U|$ is the number of users in test set, R and T represent the recommended video sequence and the true interactive video sequence, respectively. K and M denote the sizes of R and T, respectively, and $R_{ij}^{\#time}$ or $T_{ij}^{\#time}$ represents the lengths of time that user i watched video j.

5.2.2 ***Baselines.*** We implement CVRDD with multi-layer perceptron (MLP) neural networks to explore how it boosts recommendation performance and compare it with the following baselines:

- **MLP**. This method uses MLP without duration feature to model $P(Y|U, V)$ to estimate user-item matching score.
- **MLP-D** [4]. This method uses MLP with duration feature to model $P(Y|U, V, D)$ to estimate user-item matching score.
- **IPW** [30]. This method adjusts the training distribution by re-weighting training samples with propensity scores. We implement IPW with MLP.
- **DCR-MoE** [12] This method removes backdoor paths of duration by performing causal interventions with do-calculus, while modeling different duration levels with a mixture-of-experts (MoE) architecture. We implement it based on MLP for a fair comparison. MoE's hidden dimensions are searched in the range of $\{50, 100\}$.
- **Res-D2Q** [41]. This method follows the principle of backdoor adjustment and proposes a Duration-Deconfounded Quantile-based watch-time prediction framework for duration debias. We implement Res-D2Q based on MLP and discretize Wechat and ByteDance data into $L = 6$ and $L = 90$ equal-sized groups, respectively.

5.2.3 ***Hyper-parameters and training details.*** We implement the video recommendation model on WeChat using a three-layer MLP with hidden dimensions of 300, 200, and 100, respectively, and the activation function is ReLU. We optimize all models with Adam [16] optimizer with batch sizes of $\{1024, 2048\}$. We use grid search to find the optimal hyperparameters. In the CVRDD framework, α and β, which balances multi-task losses and makes the learning process of Y_{m^*} controllable, respectively, are both searched in the range of $\{0.0, 0.1, ..., 1.0\}$. The learning rate is searched in $\{1e{-}3, 1e{-}4, 1e{-}5\}$. To avoid overfitting, we set the dropout [32] to 0.2 and the patience of earlystop to 10 epochs. The reproduction code and data can be found at *https://github.com/tss-ml/cvrdd*.

5.3 Comparison

In this section, we report the recommendation performance of binary PCR and our CVRDD framework, while investigating the inference time of different models.

5.3.1 ***Different training labels.*** By analyzing the results shown in Table 2, we draw the following conclusions:

- We can obverse that almost all models perform better in binary PCR than finish playing, especially on bytedance data that contains more long videos. This can be attributed to the fact that finish playing is overly biased towards short videos at the data level. As such, models trained on such data suffer from exacerbated duration bias during training process, which makes it more difficult to debias.
- However, MLP-D is unable to achieve optimal performance in binary PCR compared to other models. We believe that MLP-D learns total causal effects of D on Y (i.e., path $D \rightarrow Y$ and $D \rightarrow M \rightarrow Y$), rendering it highly biased towards short videos. However, binary PCR changes the label distribution of finish playing, and the label reversal for long videos increases the difficulty of MLP-D training, making it unable to control the recommendation for long videos well.

5.3.2 ***Different debias schemes.*** Further, we compare the results of different models in Table 2 and draw the following conclusions:

- Our proposed CVRDD scheme with binary PCR achieves the best performance in both Wechat and ByteDance. This demonstrates the effectiveness of our scheme and its good trade-off between Top-K metrics and watch time. Such improvement can be attributed to the fact that CVRDD removes the shortcut from D to Y (i.e., path $D \rightarrow Y$) in the inference phase and leverages the indirect effect of D to Y (i.e., path $D \rightarrow M \rightarrow Y$) well.
- The poor performance of both MLP and MLP-D is because both establish spurious correlation between D and Y by the direct effect of D to Y during the training process. The reason why MLP-D outperforms MLP is that MLP-D captures the indirect effect of D on Y by modeling $P(Y|U, V, D)$, i.e., short videos are more likely to complete play.
- The reason that IPW does not show strong effectiveness is that its performance is highly dependent on the correctness of the propensity score estimates, and it suffers from high variance, leading to non-optimal results [1, 40].

Table 2: Overall top-K performance of different methods on Wechat and ByteDance. Metric@K denotes the corresponding top-K recommendation performance on this metric. FP and PCR denote finish playing and binary PCR, respectively. For each dataset, bold scores indicate the best in each column, underlined scores indicate the best baseline, and * represents the different train label for which the same model obtained the best results. For all metrics, the higher the result, the better.

	Model	Label	Recall@3	MAP@3	NDCG@3	AUTC@3	Recall@5	MAP@5	NDCG@5	AUTC@5
Wechat	MLP	FP	0.0626	0.0415	0.0456	0.0645*	0.1049	0.0554	0.0634	0.1099*
		PCR	0.0636*	0.0423*	0.0465*	0.0631	0.1081*	0.0568*	0.0649*	0.1075
	MLP-D	FP	0.0636	0.0428	0.0463	0.0632*	0.1075	0.0574	0.0649	0.1084*
		PCR	0.0666*	0.0441*	0.0477*	0.0610	0.1096*	0.0584*	0.0660*	0.1051
	IPW	FP	0.0629	0.0421	0.0461	0.0737*	0.1086	0.0569	0.0648	0.1232*
		PCR	0.0647*	0.0427*	0.0468*	0.0717	0.1088*	0.0571*	0.0651*	0.1208
	DCR-MoE	FP	0.0631	0.0419	0.0461	0.0761	0.1095	0.0573	0.0652	0.1271
		PCR	0.0665*	0.0439*	0.0478*	0.0786*	0.1099*	0.0582*	0.0659*	0.1291*
	Res-D2Q	-	0.0638	0.0433	0.0482	0.0769	0.1095	0.0583	**0.0671**	0.1280
	CVRDD	FP	0.0656	0.0444	0.0474	0.0777	0.1091	0.0592	0.0660	0.1269
		PCR	**0.0682***	**0.0462***	**0.0492***	**0.0829***	**0.1107***	**0.0601***	0.0668*	**0.1332***
	Model	Label	Recall@10	MAP@10	NDCG@10	AUTC@10	Recall@20	MAP@20	NDCG@20	AUTC@20
ByteDance	MLP	FP	0.0704	0.0232	0.0250	0.1820	0.1337	0.0290	0.0379	0.2799
		PCR	0.0725*	0.0236*	0.0252*	0.1889*	0.1368*	0.0295*	0.0382*	0.2893*
	MLP-D	FP	0.0740*	0.0250*	0.0269*	0.1731	0.1370	0.0308*	0.0398*	0.2678
		PCR	0.0734	0.0241	0.0260	0.1764*	0.1385*	0.0301	0.0393	0.2723*
	IPW	FP	0.0775	0.0254	0.0266	0.2291	0.1460	0.0316	0.0402	0.3386
		PCR	0.0810*	0.0265*	0.0278*	0.2386*	0.1502*	0.0327*	0.0415*	0.3519*
	DCR-MoE	FP	0.0785	0.0258	0.0270	0.2407	0.1479	0.0321	0.0407	0.3564
		PCR	0.0825*	0.0272*	0.0283*	0.2464*	0.1536*	0.0336*	0.0424*	0.3632*
	Res-D2Q	-	0.0785	0.0260	0.0279	0.2101	0.1473	0.0323	0.0416	0.3181
	CVRDD	FP	0.0812	0.0267	0.0279	0.2510	0.1518	0.0331	0.0420	0.3709
		PCR	**0.0899***	**0.0307***	**0.0320***	**0.2684***	**0.1631***	**0.0373***	**0.0467***	**0.3871***

Table 3: Results of ablation experiments on CVRDD. The asterisk represents the best fusion strategy $\mathcal{F}$.

	Strategy	Recall@3	MAP@3	NDCG@3	AUTC@3	Recall@5	MAP@5	NDCG@5	AUTC@5
WeChat	SUM*	**0.0682**	**0.0462**	**0.0492**	0.0829	0.1107	0.0601	**0.0668**	0.1332
	HM	0.0676	0.0455	0.0486	0.0696	**0.1117**	**0.0610**	0.0667	0.1169
	RUBI	0.0636	0.0422	0.0466	0.0727	0.1077	0.0568	0.0652	0.1220
	w/o kl	0.0659	0.0435	0.0458	**0.0920**	0.1096	0.0581	0.0641	**0.1424**
	Strategy	Recall@10	MAP@10	NDCG@10	AUTC@10	Recall@20	MAP@20	NDCG@20	AUTC@20
ByteDance	SUM*	**0.0899**	**0.0307**	**0.0320**	**0.2684**	**0.1631**	**0.0373**	**0.0467**	**0.3871**
	HM	0.0777	0.0271	0.0288	0.2301	0.1439	0.0333	0.0422	0.3269
	RUBI	0.0769	0.0253	0.0265	0.2362	0.1443	0.0314	0.0400	0.3246
	w/o kl	0.0824	0.0288	0.0291	0.2494	0.1514	0.0351	0.0437	0.3602

- Both Res-D2Q and DCR-MoE are causal intervention-based methods. Res-D2Q predicts the watch time quantile. DCR-MoE achieves the best results among the baselines. However, their approximation of the scores of the intervention terms lacks stability, resulting in the inability to obtain the best results.

5.3.3 ***Inference time.*** Table 4 shows the inference time cost on NVIDIA A100 GPU and the number of model parameters for MLP-D, DCR-MoE and CVRDD. MLP-D is the backbone of the other two schemes. The inference time cost of CVRDD is only slightly higher than MLP-D because it only adds a Res-MLP network to the backbone for modeling duration bias separately. DCR-MoE, which is the best performing baseline, suffers from significantly increased inference time cost. This is because DCR-MoE uses MoE structure to model each duration level individually, whose parameters increase with the increase of duration level L in the dataset. As can be seen, our proposed CVRDD is a light-weight and highly scalable debias scheme that achieves better performance by only making slight adjustments to the backbone.

Table 4: Comparison of model parameters and average inference time cost per batch.

Data	Wechat		ByteDance	
Model	Time(s)	Params	Time(s)	Params
MLP-D	0.06	15.0M	0.12	255.9M
DCR-MoE	0.13	15.1M	0.38	256.8M
CVRDD	0.07	15.0M	0.15	255.9M

5.4 Ablation Study

5.4.1 ***Different fusion strategies.*** The ablation results in Table 3 reveal that different fusion strategies $\mathcal{F}$ capture different direct effects on the duration bias during the training process. This indicates the summation fusion strategy outperforms the other strategies. It also shows that fusion functions with appropriate bounds can further improve the performance of CVRDD. Furthermore, the performance of the best model decreases when the restriction of KL loss is removed in training, which indicates that learning a proper Y_{m^*} is crucial for CVRDD.

Table 5: Comparison of different methods for processing Y_{m^*}

	Y_{m^*}	Recall@3	MAP@3	NDCG@3	AUTC@3	Recall@5	MAP@5	NDCG@5	AUTC@5
WeChat	learning	**0.0682**	**0.0462**	**0.0492**	0.0829	**0.1107**	**0.0601**	**0.0668**	**0.1332**
	u/v=0	0.0654	0.0432	0.0467	**0.0887**	0.1092	0.0574	0.0646	0.1319
	-	0.0637	0.0415	0.0458	0.0732	0.1075	0.0554	0.0629	0.1288
	Y_{m^*}	Recall@10	MAP@10	NDCG@10	AUTC@10	Recall@20	MAP@20	NDCG@20	AUTC@20
ByteDance	learning	**0.0899**	**0.0307**	**0.0320**	**0.2684**	**0.1631**	**0.0373**	**0.0467**	**0.3871**
	u/v=0	0.0852	0.0268	0.0296	0.2559	0.1584	0.0340	0.0481	0.3647
	-	0.0861	0.0242	0.0279	0.2571	0.1562	0.0307	0.0452	0.3754

5.4.2 ***Different learning strategies of*** Y_{m^*}. We explore three different methods of learning Y_{m^*}: **1)** learning refers to the approach used in our paper, where Y_{m^*} is set as a learnable parameter; **2)** $\mathbf{u/v = 0}$ denotes the approach which sets the user and item as zero-vectors input to f_M, i.e., $Y_{m^*} = f_M(u = 0, v = 0, d = d)$. **3)** "-" refers to the approach which ignores Y_{m^*} when modeling $Y_{m^*,d}$. We report the experimental results of the three aforementioned approaches on two real datasets in Table 5.

When using $\mathbf{u/v = 0}$ or removing Y_{m^*}, we can see a non-trivial performance drop on all metrics when compared with our **learning** approach. This denotes that using a learnable parameter Y_{m^*} is a better option. Among the three different approaches, we can see that modeling $Y_{m^*,d}$ without Y_{m^*} underperforms the other two approaches, which denotes that it is necessary to use Y_{m^*}.

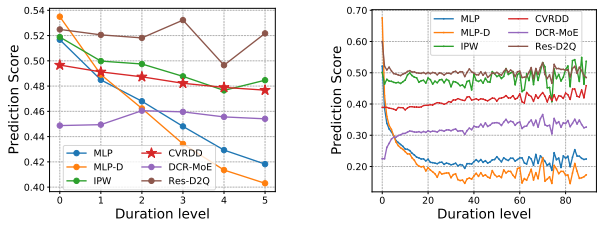
Besides, by treating Y_{m^*} as a learnable parameter rather than calculating Y_{m^*} by directly feeding $(\mathbf{u = 0, v = 0, d = d})$ into the recommendation model f_M, we see a consistent improvement on all metrics. One possible reason is that in $\mathbf{u/v = 0}$ setting, training Y_{m^*} entangled with the original recommendation model f_M might make it difficult to converge to the optimal value. However, our learning strategy, which restricts Y_{m^*} to the range of (0, 1), is more consistent with the process of human reasoning, as stated in Section 4.4.2: "In the real world, humans always tend to guess with a certain probability when it is undetermined with many treatments."

5.5 Prediction Study

In this section, we analyze the working mechanism of CVRDD in terms of prediction scores and recommendation frequencies.

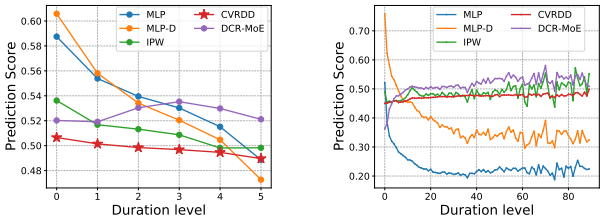
5.5.1 ***Model Prediction.*** In this subsection, we analyze the mechanisms of different models in terms of their prediction scores. Specifically, we divide the test set into L groups based on duration and calculate the average prediction scores for each group for each model. Figure 5, 6 show the average prediction scores of the models in FP and binary PCR as training labels, respectively. Figure 7 compares the average prediction scores of CVRDD under the two training labels. We have the following conclusions:

- For personalized video recommendation, the prediction score of the model is expected to change as little as possible on each duration. CVRDD has the lowest fluctuation among all curves, which means that CVRDD removes the shortcut brought by duration to the model and the prediction score obtained is a true user-item match. Also, we find that the prediction scores of CVRDD-PCR are more stable compared to CVRDD-FP, which demonstrates that our proposed binary PCR will be beneficial for model training.



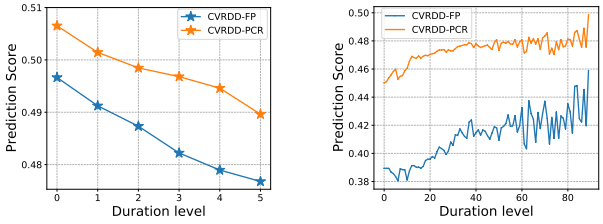
(a) Wechat (b) ByteDance

Figure 5: Model prediction scores for different durations when training label is FP.



(a) Wechat (b) ByteDance

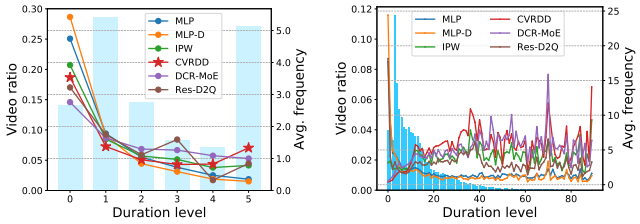
Figure 6: Model prediction scores for different durations when training label is binary PCR.



(a) Wechat (b) ByteDance

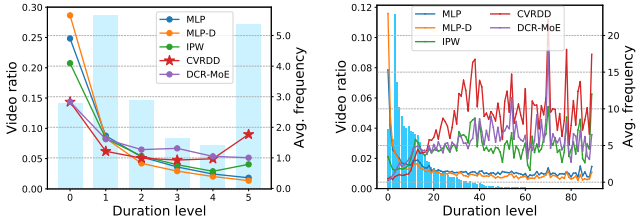
Figure 7: CVRDD prediction scores for different durations with different training labels.

- The prediction scores of MLP and MLP-D are highly correlated with duration (their curves have a significant negative correlation with duration). That is, they are heavily biased toward short videos, giving them higher scores. MLP-D explicitly models the indirect effect of D to Y while being influenced by the short connection from D to Y, so it will have higher scores in short videos compared to MLP.
- The IPW curve is flatter compared to MLP and MLP-D, but has larger fluctuations on ByteDance, suggesting that IPW can only slightly mitigate bias, which can be attributed to the fact that propensity weights are not easily estimated.



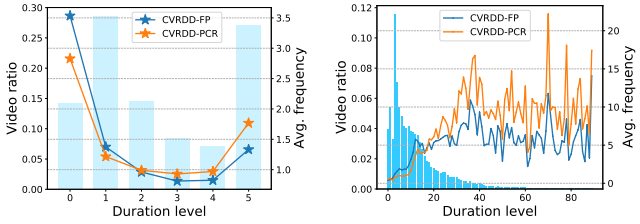
(a) Top@5 on Wechat. (b) Top@20 on ByteDance.

Figure 8: Model average recommendation frequency for different durations when training label is FP.



(a) Top@5 on Wechat. (b) Top@20 on ByteDance.

Figure 9: Model average recommendation frequency for different durations when training label is binary PCR.



(a) Top@5 on Wechat. (b) Top@20 on ByteDance.

Figure 10: CVRDD average recommendation frequency for different durations with different training labels.

- Res-D2Q uses watch duration quantile as the prediction target, and it still has a high prediction score for short videos on ByteDance, indicating that it can not eliminate duration bias well.
- The DCR-MoE is relatively flat on the curve of WeChat, however, it shows more significant fluctuations in the prediction scores of short videos on ByteDance. This indicates that DCR-MoE has a large instability in elimination bias.

5.5.2 ***Reommendation Frequency.*** In this subsection, we analyze the average recommendation frequencies of different models to further explore how CVRDD performs duration debasing. Specifically, the average recommendation frequency is the ratio of the recommendation frequency of a duration level to the number of videos in that level. Figure 8, 9 show the average recommendation frequency of the models in FP and binary PCR as training labels, respectively. Figure 10 compares the average recommendation frequency of CVRDD under the two training labels. We have the following conclusions:

- CVRDD clearly reduces the frequency of short video recommendations and increases the frequency of long video recommendations, as does DCR-MoE. In particular, Figure 10 shows us that CVRDD-PCR shows further improvement in the long and short video recommendation frequencies compared to CVRDD-FP. In contrast to FP, PCR is a loose and not significantly tendentious training label, which plays an important role in eliminating duration bias.
- As we can see, the average recommendation frequency curves of MLP and MLP-D show a long tail, indicating that they are heavily influenced by duration bias, which is not in line with the concept of personalized recommendation.
- The curve of IPW on Wechat is almost identical to that of MLP and MLP-D, which shows once again that the estimation of propensity scores is extremely unstable to the extent that it fails to achieve expectations.

6 CONCLUSION

In this paper, we study the duration bias in video recommendation. Firstly, we exploit the distributional properties of *play completion rate* to construct threshold division algorithms to obtain new training labels for alleviating the drawback of finish playing labels overly biased towards short videos. Algorithmically, we propose a model-agnostic CVRDD framework and inspect the causal relationship of video recommendation. CVRDD identifies the bad effect of duration on model prediction and eliminates it in the inference stage. It requires only a few lines of code adjustment and can be applied to any video recommendation model to achieve duration debiasing. Experiments validate that CVRDD improves top-k recommendation metrics and the average user time coverage, which is one of the metrics valued by video providers.

ACKNOWLEDGMENT

This work is supported in part by the National Key R&D Program of China under grant No. 2022YFB3105000, the National Natural Science Foundation of China under grant No. 61972189, the Shenzhen Key Lab of Software Defined Networking under grant No. ZDSYS20140509172959989, and Research Center for Computer Network (Shenzhen) Ministry of Education.

REFERENCES

[1] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240* (2020).
[2] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. 2020. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3470–3477.
[3] Andrew Collins, Dominika Tkaczyk, Akiko Aizawa, and Joeran Beel. 2018. A study of position bias in digital library recommender systems. *arXiv preprint arXiv:1802.06565* (2018).
[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
[5] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data.. In *IJCAI*. 2230–2236.
[6] Michael D Ekstrand, Mucun Tian, Mohammed R Imran Kazi, Hoda Mehrpouyan, and Daniel Kluver. 2018. Exploring author gender in book rating and recommendation. In *Proceedings of the 12th ACM conference on recommender systems*. 242–250.
[7] Weihao Gao, Xiangjun Fan, Chong Wang, Jiankai Sun, Kai Jia, Wenzhi Xiao, Ruofan Ding, Xingyan Bin, Hui Yang, and Xiaobing Liu. 2020. Deep Retrieval: Learning A Retrievable Structure for Large-Scale Recommendations. *arXiv preprint arXiv:2007.07203* (2020).

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

[9] Chunming He, Kai Li, Yachao Zhang, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. 2023. Camouflaged Object Detection with Feature Decomposition and Edge Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[10] Chunming He, Kai Li, Yachao Zhang, Guoxia Xu, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. 2023. Weakly-Supervised Concealed Object Segmentation with SAM-based Pseudo Labeling and Multi-scale Feature Grouping. *arXiv preprint arXiv:2305.11003* (2023).

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[12] Xiangnan He, Yang Zhang, Fuli Feng, Chonggang Song, Lingling Yi, Guohui Ling, and Yongdong Zhang. 2022. Addressing Confounding Feature Issue for Causal Recommendation. *arXiv preprint arXiv:2205.06532* (2022).

[13] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *International Conference on Machine Learning*. PMLR, 1512–1520.

[14] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately interpreting clickthrough data as implicit feedback. In *Acm Sigir Forum*, Vol. 51. Acm New York, NY, USA, 4–11.

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

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

[17] Sanjay Krishnan, Jay Patel, Michael J Franklin, and Ken Goldberg. 2014. A methodology for learning, analyzing, and mitigating social influence bias in recommender systems. In *Proceedings of the 8th ACM Conference on Recommender systems*. 137–144.

[18] Gael Lederrey and Robert West. 2018. When sheep shop: measuring herding effects in product ratings with natural experiments. In *Proceedings of the 2018 World Wide Web Conference*. 793–802.

[19] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI. AUAI*.

[20] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2019. Crank up the volume: preference bias amplification in collaborative recommendation. *arXiv preprint arXiv:1909.06362* (2019).

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

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

[23] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12700–12710.

[24] Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* 9, 1 (1979), 62–66.

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

[26] Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *The World Wide Web Conference*. 1443–1453.

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

[28] Judea Pearl. 2022. Direct and indirect effects. In *Probabilistic and Causal Inference: The Works of Judea Pearl*. 373–392.

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

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

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

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[33] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems*. 213–220.

[34] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1717–1725.

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

[36] Xin Wang, Steven CH Hoi, Martin Ester, Jiajun Bu, and Chun Chen. 2017. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International Conference on World Wide Web*. 1601–1610.

[37] Xiangmeng Wang, Qian Li, Dianer Yu, Peng Cui, Zhichao Wang, and Guandong Xu. 2022. Causal Disentanglement for Semantics-Aware Intent Learning in Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[38] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1791–1800.

[39] Hongyi Wen, Longqi Yang, and Deborah Estrin. 2019. Leveraging post-click feedback for content recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 278–286.

[40] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 5 (2021), 1–46.

[41] Ruohan Zhan, Changhua Pei, Qiang Su, Jianfeng Wen, Xueliang Wang, Guanyu Mu, Dong Zheng, Peng Jiang, and Kun Gai. 2022. Deconfounding Duration Bias in Watch-time Prediction for Video Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4472–4481.

[42] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 11–20.