Alleviating Confounding Effects with Contrastive Learning in Recommendation

Di You[1] and Kyumin Lee[1]

Worcester Polytechnic Institute, Worcester MA, USA
{dyou,kmlee}@wpi.edu

Abstract. Recently, there has been a growing interest in mitigating the bias effects in recommendations using causal inference. However, Rubin's potential outcome framework may produce inaccurate estimates in real-world scenarios due to the presence of hidden confounders. In addition, existing works adopting the Pearl causal graph framework tend to focus on specific types of bias (e.g., *selection bias*, *popularity bias*, *exposure bias*) instead of directly mitigating the impact of hidden confounders. Motivated by the aforementioned limitations, in this paper, we formulate the recommendation task as a causal graph with unobserved/unmeasurable confounders. We present a novel causality-based architecture called Multi-behavior Debiased Contrastive Collaborative Filtering (MDCCL) and apply the front-door adjustment for intervention. We leverage a pre-like behavior such as *clicking an item* (i.e., a behavior occurred before the target behavior such as *purchasing*) to mitigate the bias effects. Additionally, we design a contrastive loss that also provides a debiasing effect benefiting the recommendation. An empirical study on three real-world datasets validates that our proposed method successfully outperforms nine state-of-the-art baselines. Code and the datasets will be available at *https://github.com/queenjocey/MDCCL*.

Keywords: Recommender system, causal inference, debiased recommendation, contrastive learning

1 Introduction

While we have witnessed the success of recommender systems in various domains (e.g., social platform [41], music sites [29], e-commerce platforms [30]), most recommendation models focus on fitting observed user-item historical interactions [10, 11, 15, 24]. However, user-item interaction data, which forms the basis for training recommendation models, are observational rather than experimental [3]. A common practice in training conventional recommender systems is to consider the unobserved interactions as negative feedback, assuming the observed data are missing-at-random. However, user interaction data are always missing-not-at-random [19, 27, 40] in reality. While the matching-based method models the correlation between a user and candidate items, it does not inherently reflect the true causal relationship between user-item interaction. The presence of confounders, such as item quality, can lead to misleading recommendation results. To demonstrate the impact of these confounders, we present a toy example

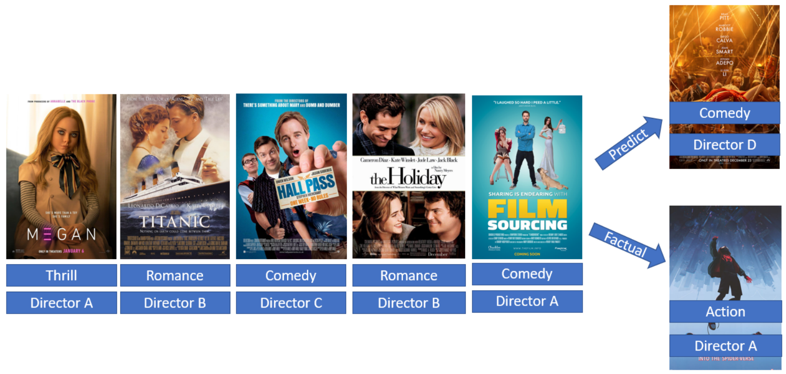


Fig. 1: A toy example where confounders mislead the recommendation result.

in Fig. 1. The user's historical interactions on the left suggest a preference for comedy based on genre features. However, in this specific case, the user's choice of next movie is influenced by the director rather than the genre. Although there exists a strong correlation between the genre feature and user preference, genre serves as a confounder here, and the next consumed movie of the target user is actually driven by causation. Meanwhile, confounders induce bias, which leads to the bias effects in the correlations estimated from the observations.

There has been a surge of study exploring bias elimination in recent years [32, 33,37,39,43,45]. For example, the inverse propensity score (IPS)-based approach has long been popular in the recommendation community [27,42], however, this line of work heavily relies on the estimation of the IPS score and suffers high variance issues. Another line of work formulates their research with a causal graph to describe causal relationships and conducts reasoning over the graph to estimate causal effect [33, 37]. Existing works [33, 37, 43, 49] adopting the Pearl causal graph framework mostly follow the following pattern: (1) identify a specific confounder first, and (2) propose a confounder-aware model to address the specific confounder. However, in real-world scenarios, it is unrealistic to identify specific confounders. Moreover, not all confounders are observable and measurable, which limits the efficacy of the existing tools. Fortunately, there is another tool named front-door adjustment [23], which allows us to deal with any types of confounders, including unobservable/unmeasurable confounders.

Motivated by the aforementioned analysis, we design a new causal graph as shown in Fig. 3 where unobserved/unmeasurable confounders exist with a mediator node. Then, we propose a novel Multi-behavior Debiased Contrastive Collaborative Filtering (**MDCCL**) framework that leverages the front-door adjustment to eliminate the bias effect induced by confounders. As for the choice of mediator, we utilize prior user feedback about items (i.e., click an item) to facilitate unbiased recommendation, which formulates our task as multi-behavior recommendation. Further, we design a debiased contrastive loss component to mitigate bias effect and improve recommendation accuracy. In the experiments, we compare our model with nine state-of-the-art baselines. Empirical experi-

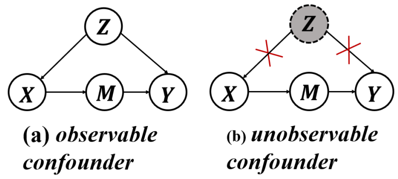


Fig. 2: X, Y, M and Z represent treatment variable, outcome variable, mediator variable and confounder variable, respectively. Figure (a) shows Z as observed and measurable, while in Figure (b), Z is unobserved or unmeasurable.

ments and in-depth analysis validate the effectiveness of MDCCL algorithm on both accurate recommendation and deconfounding.

2 Preliminaries

2.1 Task Formulation

U=$\{u_1, u_2, ..., u_k\}$ as a set of all users where $k = |U|$ is the total number of users, and P=$\{p_1, p_2, ..., p_n\}$ as a set of all items where $n = |P|$ is the total number of items. Without loss of generality, we assume that the number of behaviors is T, and we use $Y_1, Y_2, ..., Y_T$ denotes behavior matrices if the user interacted with an item under behavior t, where $Y_1, Y_2, ..., Y_{T-1}$are auxiliary behaviors, and Y_T is the target behavior. We consider interaction matrices in the binary form, in which each entry has value 1 if user u and item p interacted under behavior t, otherwise value 0. To simplify our analysis, in our work, we mainly discuss the target behavior, denoting as y and click behavior, serves as mediator, denoting as m. Bold versions of those variables, which we will introduce in the following sections, indicate their respective latent representations/embeddings.

2.2 Preliminaries on Causal Inference

In this section, we will briefly introduce some basic concepts and theorems in causal inference.

Definition 1: Causal Graph. Causal graph is a directed acyclic graph(DAG), where $G = (\mathcal{N}, \mathcal{E})$, describing the causal relationship. $\mathcal{N}$ represents a set of nodes, containing variables in U and P in recommendation; and $\mathcal{E}$ represents a set of edges, also known as the causal relations.

Definition 2: Backdoor adjustment [23]. Given an ordered pair of variables (X, Y) in a causal graph G, a set of variables Z satisfies the back-door criterion with respect to (X, Y) if Z satisfies the following conditions:

- No node in Z is a descendant of X;
- Z blocks every path between X and Y that contains an arrow into X.

With the help of a set of variables that satisfy the back-door criterion, we can adjust the effect of measured confounders. We take the causal graph in Fig. 2(a) as an example. Considering the treatment variable X and the outcome variable Y, we want to estimate the effect of X on Y, denoted as $P(Y = y|do(X = x))$.

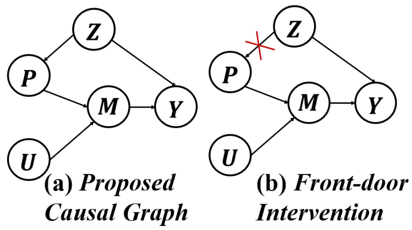


Fig. 3: Causal Graph of (a) the proposed model and (b) front-door intervention.

Due to the existence of confounder Z (i.e., Z is a parent node of X), we cannot conclude that $P(Y=y|do(X=x)) = P(Y=y|X=x)$. However, since variable Z satisfies the back-door criterion, we use it to adjust the effect, in other words, we are accounting for and measuring all confounders [23]. Therefore, we compute

$$P(Y=y|do(X=x)) = \sum_{z} P(Y=y|X=x, Z=z)P(Z=z) \tag{1}$$

However, a serious limitation is that the above equation assumes that the confounder variables are all measurable and satisfy the backdoor criterion. However, unobservable and hidden confounders always exist in recommender systems [4, 47]. Given this setting, we introduce the front-door criterion.

Definition 3: Frontdoor criterion and adjustment. Given an ordered pair of variables (X, Y) in a causal graph G, a set of variables M satisfies the front-door criterion with respect to (X, Y) if (X, Y) satisfies the following conditions:

- M intercepts all directed paths from X to Y;
- There is no unblocked path from X to M;
- X blocks all back-door paths from M to Y

If a set of variables M satisfies the front-door criterion related to an ordered pair of variables (X, Y), and if $P(x, z) > 0$, then the causal effect of X on Y is identifiable and is given by

$$P(y|do(x)) = \sum_{m} P(m|x) \sum_{x'} P(y|x', m)P(x') \tag{2}$$

We take Fig. 2(b) as an example. In this case, the variable Z here is not measurable so that back-door adjustment cannot be directly applied. However, it satisfies the front-door criterion, allowing us to utilize the front-door adjustment to handle the unmeasurable confounder Z. Intuitively, the desired effect can be expressed as follows

$$P(y|do(x)) = \sum_{m} P(m|do(x))P(y|do(m)) \tag{3}$$

3 Methodology

3.1 Causal view of Deconfounding Recommendation

Fig. 3 shows our proposed causal graph for interaction generation when the confounding feature exists. Next, we explain the semantics of the causal graph.

- Node U and P denote the user and item ID embeddings, respectively. In this work we only use ID feature.
- Node Z denotes the hidden confounder, which can be generated for various reasons (e.g., the producer's motivation, and the item's quality). Notice that the hidden confounder in our discussion is either unobservable or unmeasurable, thus the backdoor adjustment is not applicable.
- Node M denotes the mediator, which is pre-like/pre-purchase behavior (i.e., *click an item*) in this work. Node Y denotes the target interaction label (i.e., *purchase* or *like*);
- Edge $P \leftarrow Z \rightarrow Y$ denotes that hidden confounder Z affect both item features and happening of interaction, while it does not necessarily reflect users' real preference;
- Edge $\{P, U\} \rightarrow M \rightarrow Y$ denotes that user preference and item features jointly determine the level of user-item matching. And based on the front-door criterion, we can also observe that the prior feedback functions as a mediator in our task formulation (e.g., click $\rightarrow$ like on micro-video platforms.)

Most of conventional recommendation algorithms directly estimate the correlation $P(Y|U,P)$ using historical interaction data, which leads to biased estimation for recommended results. While previous causal models estimate the causal effect $P(Y|U,do(P))$, they overlook the effect of hidden confounder Z, thus bias issue still exists. In our work, we propose to simultaneously cut off the direct effect of $Z \rightarrow P$ and backdoor path $P \leftarrow Z \rightarrow Y$ to eliminate the confounding effect in our estimation.

3.2 Multi-behavior Debiased Contrastive CF

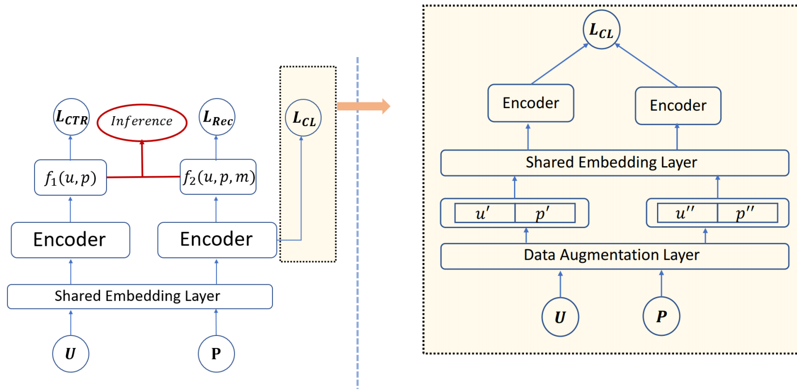
In this section, we discuss how to mitigate the confounding effect without measuring the confounder Z. We introduce prior feedback (i.e., *click*) as mediator as shown in Fig. 4.

Intervention with do-calculus. Considering that the hidden confounder Z is unobservable or unmeasurable, we apply the front-door adjustment tool for user's preference estimation towards items. Specifically, we estimate the distribution as follows:

$$\begin{aligned} P(y|u,do(p)) &= \sum_m P(m|u,do(p)) \sum_z P(z)P(y|u,z,m) \\ &= \sum_m P(m|u,do(p))P(y|u,do(m)) \end{aligned} \tag{4}$$

In Eq. 4, the first term denotes the probability of mediator M being a set as m given certain item features, which reflects the causal effect of P on M. On the other hand, the second term denotes the probability of y when m happens, which is the causal effect of M on Y. The equation holds because of the backdoor criterion. Fortunately, both terms are measurable in our formulation.

Estimating $P(m|u,do(p))$. $P(m|u,do(p)) = P(m|u,p)$ since the backdoor path $P \leftarrow Z \rightarrow Y \leftarrow M$ is $d-$separated by collider Y, which means given Y, Z is inde-



(a) MDCCL's overall architecture (b) Contrastive Learning Structure

Fig. 4: Our proposed architecture where in Fig. 4a blue arrows indicate the training stage and the red arrows represent the inference stage.

pendent of another set M. *Estimating* $P(y|u,do(m))$ According to [23], blocking Z is equivalent to blocking P in the backdoor path $M \leftarrow P \leftarrow Z \rightarrow Y$. Also, according to our causal graph, M is independent of Z given P, and P is independent of Y given Z and M. Therefore, following prior causal recommendation work [43, 47], we can apply backdoor adjustment as follows:

$$\begin{aligned} P(y|u,do(m)) &= \sum_z P(z)P(y|z,m,u) \\ &\overset{\text{(a)}}{=} \sum_z \sum_p P(z|p)P(p)P(y|z,m,u) \\ &\overset{\text{(b)}}{=} \sum_p \sum_z P(z|p)P(p)P(y|z,m,u,p) \\ &\overset{\text{(c)}}{=} \sum_p (\sum_z P(y|m,z,u,p)P(z|p,m))P(p) \\ &\overset{\text{(d)}}{=} \sum_p P(y|u,p,m)P(p) \end{aligned} \tag{5}$$

We illustrate the derivation steps as follows:

- (a) holds due to $P(z) = P(z|p)P(p)$
- (b) holds since P is independent of Y given Z and M, thus we have $P(y|z,m,u) = P(y|z,m,u,p)$
- (c) is induced by $P(z|p) = P(z|p,m)$, since M is independent of Z given P
- (d) holds because of marginal distribution properties

Therefore, we derived the following equation to replace Eq. 4:

$$P(y|u,do(p)) = \sum_m P(m|u,p) \sum_{p'} P(y|u,p',m)P(p') \tag{6}$$

where we can get rid of $P(p')$ safely. Based on the above analysis, we present our deconfounding architecture in two stages: training and inference.

Deconfounded Training. In the training stage, as it showed in Fig. 4a, we shall estimate probability $P(m|u,p)$ and $P(y|u,p,m)$.

Since modeling $P(m|u,p)$ is equivalent to the well-known CTR prediction task, we parameterized it as $f_1(u,p)$, where $f_1(\cdot)$ can be any backbone model. On the other hand, $P(y|u,p,m)$ estimation is our main recommendation task, and can be decomposed into a late-fusion manner [31, 33] without loss of generality,

$$f_2(u,p,m) = f_2'(u,m) * \sigma(f_2''(u,p)) \tag{7}$$

, where f_2' and f_2'' are both backbone encoders, $\sigma(\cdot)$ is sigmoid function that introduces non-linearity for sufficient representation capacity of the fusion strategy. As it showed in Fig. 4a, we can use any existing model as backbone encoders to model in our framework. For simplicity, we adopt LightGCN for all components and take only the ID features of users and items as inputs.

The backbone model has different target values so that we estimate $P(m|u,p)$ and $P(y|u,p,m)$ as following:

$$\mathcal{L}_{CTR}(\mathcal{D}|\Phi) = -\sum_{(i,j^+,j^-)} log\sigma(s_{ij^+} - s_{ij^-}) + \lambda_\Phi \|\Phi\|_2 \tag{8}$$

$$\mathcal{L}_{Rec}(\mathcal{D}|\Theta) = -\sum_{(i,j^+,j^-)} log\sigma(o_{ij^+} - o_{ij^-}) + \lambda_\Theta \|\Theta\|_2 \tag{9}$$

, where $(i,\, j^+,\, j^-)$ is a triplet of a target user, a positive item, and a negative item that is randomly sampled from the items set P. $\mathcal{D}$ denotes all the training instances. s_{ij^+} and s_{ij^-} are the respective positive and negative preference scores in CTR task. o_{ij^+} and o_{ij^-} are the respective positive and negative preference scores in the recommendation task. Φ and Θ are trainable parameters, and λ_Φ and λ_Θ are hyperparameters for regularization terms.

Contrastive Learning. We enhance robustness in learned representations by employing contrastive learning and data augmentation. A challenging procedure to apply contrastive learning in recommendation is to compose positive and negative pairs. We create a user-item bipartite graph and generate correlated views for each node, whether a user or an item, with its neighbor nodes under different behavior types by incorporating adaptive data augmentation techniques such as node drop. This approach intentionally reduces the impact of popular nodes while preserving isolated node information, mitigating popularity bias. These generated views are then input into the backbone model, treating views from the same node as positive pairs and views from different nodes as negative pairs. Furthermore, building upon recent work [36] on adaptive edge and node dropping, we adopt the idea of principle. This principle encourages representations to capture only the necessary information for the downstream task, minimizing mutual information between the original graph and generated views while maintaining recommendation performance.

To estimate mutual information between augmentation views, which encompasses both user and item perspectives, we utilize negative InfoNCE, as suggested by [8, 21], which is equivalent to maximizing the lower bound of mutual information. We formally define our contrastive loss for representations as:

$$\mathcal{L}_{CL_u} = -\log \frac{\exp\left(s(u_i', u_i'')/\tau\right)}{\sum_{i'}^{k} \exp\left(s(u_i', u_{i'}'')/\tau\right)}, \mathcal{L}_{CL_p} = -\log \frac{\exp\left(s(p_j', p_j'')/\tau\right)}{\sum_{j'}^{n} \exp\left(s(p_j', p_{j'}'')/\tau\right)} \quad (10)$$

where $\mathcal{L}_{CL_u}$ and $\mathcal{L}_{CL_p}$ are contrastive loss for user and item,respectively. $s(\cdot)$ denotes the cosine similarity function, and τ is the tunable temperature hyperparameter to adjust the scale for softmax. (u_i', u_i'') and $(u_i', u_{i'}'')$ are positive and negative user pairs, respectively. Similarly, (p_j', p_j'') and $(p_j', p_{j'}'')$ are positive and negative item pairs, respectively.

Formally, we incorporate the contrastive loss into our training schema as:

$$\mathcal{L}_{Total} = \mathcal{L}_{Rec} + \alpha * \mathcal{L}_{CTR} + \beta * (\mathcal{L}_{CL_u} + \mathcal{L}_{CL_p}) \quad (11)$$

, where α and β are hyperparameters controlling the effect of auxiliary tasks.

Inference. At the inference stage, we estimate the causal effect of the user-item pair with Eq. 6 and adopt a fusion strategy:

$$\begin{aligned} P(y|u, do(p)) &= \sum_m P(m|u,p) \sum_p^{\prime} P(p')P(y|u,p',m) \\ &= \sum_m f_1(u,p) * \sum_{p'} f_2(u,p',m)P(p') \\ &= \sum_m f_1(u,p) f_2'(u,m) \sum_{p'} f_2''(u,p')P(p') \\ &\propto \sum_m f_1(u,p) f_2'(u,m) \sum_{p'} f_2''(u,p') \end{aligned} \quad (12)$$

Notice that $\sum_{p'} f_2''(u,p')$ is a constant value given u that can be omitted, thus Eq. 12 reduces to $P(y|u, do(p)) = \sum_m f_1(u,p) f_2'(u,m)$

3.3 Two Assumptions in Our Framework

In our proposed framework, we make two assumptions: Firstly, we focus on confounders positioned between users' click and like/purchase behaviors, such as item quality in the KuaiRec dataset(one of three datasets that we used), which encompasses various aspects, including resolution, content, and interestingness. Essentially, our introduced mediator remains independent of the influence of these confounders. Second, we acknowledge that confounders may affect auxiliary behaviors to varying degrees, but we simplify this assumption in our initial attempt to address confounder effects in multi-behavior recommendation, as articulated above. In future research, we will explore the impact of different confounders on various behaviors, which is beyond our current scope.

Table 1: Statistics of datasets.

Dataset	|Users|	|Items|	|Clicks|	| Likes/Purchases|	Overall Density(%)	Duration
Fliggy	2,730,201	104,342	32,444,647	1,160,723	0.01%	6 month
KuaiRec	7,176	10,729	12,530,806	1,124,378	17%	2 month
Adressa	31,123	4,895	1,437,540	998,612	1.59%	1 week

4 Empirical Study

4.1 Experimental Setup

Datasets. We evaluate all models on **three** public benchmark datasets collected from three real-world systems:

- *Fliggy Dataset* [28] is extracted from users' behavior logs at Fliggy in 2021, a prominent Chinese online travel portal Among various user behaviors, we only use *click* and *buy* to keep consistency with the other datasets.
- *KuaiRec Dataset* [5] is a dataset collected from the logs of the Kuaishou video-sharing mobile app. It features a "fully observed" user-item interaction matrix, minimizing missing values as each user has interacted with every video and provided feedback. Given the absence of explicit "like" behavior, we adopt the approach outlined in [5], considering a click with a **watch-ratio = play-duration/video-duration** $>$ **2** as "like" behavior.
- *Adressa Dataset* [6] is a news dataset. Following the prior study [13], we treat a click with dwell time > 30 seconds as "like" behavior.

The detailed information about the datasets is presented in Table 1.

Evaluation Protocol and Metrics. For data preprocessing, we adopted a popular *k-core* preprocessing step [9] (with k=5), filtering out users and items with less than 5 interactions.

Following the prior works [17, 30], each dataset is sorted by timestamp, and split to train/valid/test sets with corresponding 70%/10%/20% proportions. We used the same split for our model and baselines for a fair comparison. For evaluation, we followed [34, 38] to sample 1,000 unobserved items, with which a user did not interact before a specific target behavior, considering them as negative items. Finally, we used them along with all positive items in the test set. We adopted Recall and NDCG as evaluation metrics.

Compared baselines. We compared our proposed model with nine state-of-the-art recommendation models as follows:

- **MF-BPR** [24]: MF-BPR is a widely-used collaborative filtering baseline optimized by Bayesian personalized ranking (BPR) loss.
- **LightGCN** [10]: It is a graph neural network model that simplifies the original design of GCN so that it can fit better to recommendation applications.
- **IPW** [27]: It adds the standard Inverse Propensity Weight to reweight samples to alleviate item popularity bias.

- **Multi-DR** [42]: It uses Multi-task Inverse Propensity Weighting (Multi-IPW) estimator and Multi-task Doubly Robust (Multi-DR) estimator to mitigate selection bias and data sparsity in multi-behavior recommendation.
- **MACR** [37]: It is model-agnostic using a counterfactual reasoning method for eliminating popularity bias.
- **CR** [33]: It is a counterfactual inference-based method that addresses the clickbait issue. CR aims to capture unbiased user preferences without using like feedback. We use code released by the authors for implementation, using MMGCN as backbone. We use code released by the authors to reimplement experiments, where CR is also implemented based on MMGCN and takes exposure features as input.
- **PDA** [43]: It is a state-of-the-art method that performs de-confounded training while intervening the popularity bias during model inference. The authors provide two versions, where PD directly uses matching score for recommendation and PDA leverages predicted item popularity score in recommendation. We adopt the popularity-adjusted version in our work.
- **RD-DR** [4] accounts for the effect of unmeasured confounders on propensities, under the mild assumption that the effect is bounded.
- **HCR** [47] propose to leverages front-door adjustment to decompose the causal effect into two partial effects, which are independent from the hidden confounder and identifiable.

4.2 Implementation Details

We thoroughly tuned the baselines' hyperparameters to achieve optimal performance on the validation set. Early-stop training strategy was applied based on Recall@20 and NDCG@20 on the validation set with a patience of 20 epochs. We performed a grid search of the latent dimension size in the range of $\{16, 32, 64, 128\}$, the regularization weights (λ_Φ and λ_Θ) in the range of $\{0.00001, 0.0001, 0.001, 0.01\}$, and the learning rate in the range of $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$. We train all models with Adam optimizer [14]. After performing hyperparameter search, the learning rate was set to 0.0005, the batch size was set to 1024, and the size of the latent factor was set to 128. We tune the hyperparameters α and β in the range of $\{0.2, 0.4, 0.6, 0.8\}$ and ensures that $\alpha + \beta = 1$. The optimal combination of loss weight achieved with the combination of (CTR loss, CL loss) = (0.6, 0.4), indicating the importance of both auxiliary tasks.

4.3 Overall Performance and Ablation Study

The recommendation performance of baselines and our model is shown in Table 2. Among all baselines, bias-aware baselines perform better than conventional baselines in general. Remarkably, our MDCCL consistently outperformed all baselines in all three datasets. We underlined the best baseline. On average, our proposed model improved 9.72% at Recall@10 and 6.72% at NDCG@10 compared with the best baseline. A similar trend also appears at Top-20.

Table 2: Overall performance at top-10 on three real-world datasets. The best performance is in bold, the best baseline result is underlined. The last column shows relative improvement of our MDCCL over the best baseline.

		MF	LightGCN	IPW	Multi-DR	MACR	CR	PDA	RD-DR	HCR	**MDCCL**	**Imp. %**
KuaiRec	Recall@10	0.0054	0.0142	0.0108	0.0145	0.0239	0.0298	0.0264	0.0279	0.0312	**0.0332**	**6.41%**
	NDCG@10	0.0041	0.0113	0.0077	0.0127	0.0142	0.0279	0.0231	0.0261	0.0301	**0.0320**	**6.31%**
Adressa	Recall@10	0.0642	0.1034	0.0804	0.0919	0.124	0.1452	0.1021	0.1399	0.1573	**0.1771**	**12.58%**
	NDCG@10	0.0453	0.0794	0.0663	0.0817	0.1074	0.1078	0.0997	0.1126	0.1176	**0.1246**	**5.95%**
Fliggy	Recall@10	0.3712	0.3951	0.3862	0.4011	0.4295	0.4077	0.4023	0.4277	0.4261	**0.4732**	**10.17%**
	NDCG@10	0.1807	0.2077	0.1974	0.2115	0.2459	0.243	0.2015	0.2391	0.2442	**0.2653**	**7.89%**

Table 3: Ablation analysis.

	KuaiRec		**Adressa**		**Fliggy**	
	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
MDCCL	***0.0332***	***0.032***	***0.1771***	***0.1246***	***0.4732***	***0.2653***
MDCCL w/o MB	0.0242	0.0217	0.1154	0.0832	0.4153	0.2273
MDCCL w/o CL	0.0295	0.0274	0.1477	0.0944	0.4425	0.2346
MDCCL w/o DA	0.0301	0.0293	0.1582	0.1021	0.4593	0.2395
LightGCN(Backbone model)	0.0142	0.0113	0.1034	0.0794	0.3951	0.2077

We observe that both IPW-based methods (*IPW*, *Multi-DR*) perform worse than other causal recommender models (*MACR, CR, PDA, RD-DR, HCR*). We postulate that this line of methods heavily relies on estimating a proper propensity score, which is non-trivial and typically suffers from high variance. *Multi-DR*, which augmented its architecture with an additional imputation model for robustness achieves better results. The counterfactual world constructed by *MACR* directly removes all natural direct effect from both the user-side and item-side regardless of whether it is harmful or not, which leads to its compromised accuracy. *PDA* models users' interest drift across time for bias adjustment, and turns to be effective for debiasing. *CR* achieves a competitive performance in most cases indicating that the clickbait issue has been a serious obstacle to produce accurate recommendation. *RD-DR* and *HCR* achieve impressive results indicating that hidden confounders impact on the recommendation quality.

To verify the effectiveness of each design in our framework, we developed three variants and summarize results in Table 3:

- **MDCCL w/o contrastive learning (CL):** We remove the contrastive learning while keeping deconfounded recommendation framework intact.
- **MDCCL w/o data augmentation (DA):** We use original subgraphs in contrastive learning without applying any data augmentation.
- **MDCCL w/o multi-behavior modeling (MB):** We set the α to 0, which disables auxiliary behavior modeling $f_1(u, p)$ – CTR – branch while keeping contrastive learning as an auxiliary task.

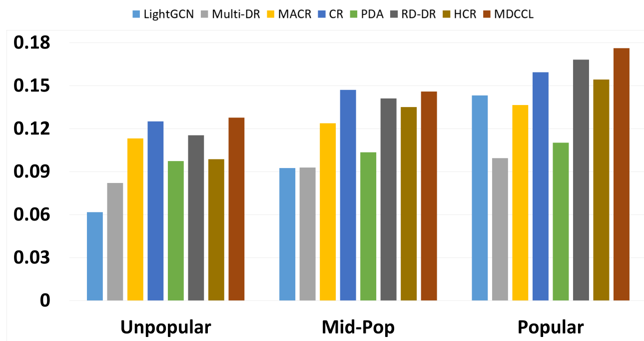


Fig. 5: Recall@10 at Adressa dataset grouped by popularity.

We can observe that the aforementioned designs all contribute positively to our proposed MDCCL, however, the importance varies greatly. Data augmentation (DA) contributes the least and degrades performance in one case. It makes sense because DA neither removes confounding features nor changes the modeling process, so inappropriate data augmentation may introduce noise and thus negatively affects the model prediction. Contrastive learning helps to further differentiate between similar items and learn more robust representations. Multi-behavior modeling makes the most contribution as it is the key component of the debiasing strategy in our proposed framework. The performance in Table 3 further proves the effectiveness of our proposed MDCCL.

4.4 Debiasing Effect

We analyze the effectiveness of our model against biases via mitigating the impact of hidden confounders. In particular, as exemplars/case studies, we mainly show effectiveness of our model against popularity bias and exposure bias. However, the effectiveness is not limited to these two biases because our work focused on the hidden confounders.

Popularity bias. We gauge popularity of each item using D_i/D_{total}, where D_i denotes the number of interactions an item involved, and D_{total} is the total interactions in each training set. Sorting items by popularity, we split the dataset into three subsets: unpopular, mid-pop, and popular, ensuring equal total popularity across subsets (Fig. 5). We evaluate Recall@10 by item popularity, expecting similar performance since the sum of popularity in each subset is equal. Remarkably, *MDCCL* consistently outperforms or competes well across all datasets, mitigating popularity bias effectively. Due to space constraints, we visualize the Adressa dataset only. In summary, all models excel on the popular subset, but conventional recommender *LightGCN* performs much worse on the unpopular subset compared to our model, which excels consistently in all subsets, indicating effective popularity bias mitigation.

Exposure bias. In this experiment, we assess pretrained models under a test set without exposure bias. Fortunately, [5] provides a fully observed KuaiRec

Table 4: Performance of models on a test set without exposure bias.

		LightGCN	IPW	Multi-DR	MACR	CR	PDA	RD-DR	HCR	**MDCCL**	**Imp. %**
KuaiRec (no exposure bias)	Recall@10	0.0119	0.0105	0.0145	0.0272	0.031	0.0262	0.0316	0.0311	**0.0335**	**6.01%**
	NDCG@10	0.0092	0.0072	0.0121	0.0154	0.0291	0.0225	0.029	0.0294	**0.0312**	**6.12%**

small matrix for testing, ensuring over a 99% exposure rate. It's important to note that there is no interaction overlap between the KuaiRec training and validation sets, and this test set. The experiment results in Table 4 demonstrate our model's continued superiority over baselines. This indicates the model's capacity to mitigate exposure bias influence and bolster recommendation robustness, although our model was training on data with various biases beyond exposure bias and did not exclusively focus on exposure bias.

5 Related Work

Researchers have addressed bias through methods like debiasing techniques. To name a few, [3] categorized common biases into seven types, with selection bias [12, 27], exposure bias [18, 20, 22, 48] , and popularity bias [37, 43, 44, 46] being the most discussed. Existing approaches have either relied on heuristic rules [1,16,26] or have been sensitive to pseudo-labels for data imputation [25,35]. To mitigate bias in recommendation, IPS-based approaches [7, 27, 35], often combined with data imputation, were proposed and became popular. However, improper propensity scores can lead to inaccuracies and high variance. More recently, causality-based methods have been introduced to provide more accurate and explainable solutions. [39, 43] involved causal intervention to remove bias factors from inference via $do-calculus$. Knowledge distillation methods [2] train a teacher model on the uniform dataset and then use it to guide the base model trained on biased dataset. Counterfactual-based methods [33, 37] estimated causal effects by comparing the factual world with the counterfactual world, targeting different confounding features. Unlike the prior works, we have proposed leveraging a pre-like behavior such as click an item as the mediator to mitigate the bias effects caused by confounders in recommendation systems.

6 Conclusion

In this paper, we analyze confounding effect in recommender systems from the perspective of causal graph. Considering confounders in real-world scenarios are much more complex than assumptions in existing work. Therefore, we propose to utilize users' prior feedback as mediator and apply the front-door adjustment to free the influence of unobserved confounders from inference. Further, we develop an auxiliary contrastive learning task to ensure the robustness of learned representation. Extensive experiments prove the effectiveness of our model in both accuracy and estimating confounding effects.

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Controlling popularity bias in learning-to-rank recommendation. In: Recsys. pp. 42–46 (2017)

2. Chen, J., Dong, H., Qiu, Y., He, X., Xin, X., Chen, L., Lin, G., Yang, K.: Autodebias: Learning to debias for recommendation. In: SIGIR. pp. 21–30 (2021)
3. Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., He, X.: Bias and debias in recommender system: A survey and future directions. ACM Transactions on Information Systems (2020)
4. Ding, S., Wu, P., Feng, F., Wang, Y., He, X., Liao, Y., Zhang, Y.: Addressing unmeasured confounder for recommendation with sensitivity analysis. In: KDD. p. 305–315 (2022)
5. Gao, C., Li, S., Lei, W., Chen, J., Li, B., Jiang, P., He, X., Mao, J., Chua, T.S.: Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In: CIKM (2022)
6. Gulla, J.A., Zhang, L., Liu, P., Özgöbek, O., Su, X.: The adressa dataset for news recommendation. In: WI (2017)
7. Guo, S., Zou, L., Liu, Y., Ye, W., Cheng, S., Wang, S., Chen, H., Yin, D., Chang, Y.: Enhanced doubly robust learning for debiasing post-click conversion rate estimation. In: SIGIR (2021)
8. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: AISTATS. pp. 297–304. JMLR Workshop and Conference Proceedings (2010)
9. He, R., McAuley, J.: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: WWW. pp. 507–517 (2016)
10. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR. pp. 639–648 (2020)
11. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW. pp. 173–182 (2017)
12. Hernández-Lobato, J.M., Houlsby, N., Ghahramani, Z.: Probabilistic matrix factorization with non-random missing data. In: ICML (2014)
13. Kim, Y., Hassan, A., White, R.W., Zitouni, I.: Modeling dwell time to predict click-level satisfaction. In: WSDM. pp. 193–202 (2014)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) ICLR, (2015)
15. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
16. Li, Y., Hu, J., Zhai, C., Chen, Y.: Improving one-class collaborative filtering by incorporating rich user information. In: CIKM (2010)
17. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: WWW. pp. 689–698 (2018)
18. Liu, D., Cheng, P., Zhu, H., Dong, Z., He, X., Pan, W., Ming, Z.: Mitigating confounding bias in recommendation via information bottleneck. In: Recys (2021)
19. Liu, D., Lin, C., Zhang, Z., Xiao, Y., Tong, H.: Spiral of silence in recommender systems. In: WSDM. pp. 222–230 (2019)
20. Ma, X., Zhao, L., Huang, G., Wang, Z., Hu, Z., Zhu, X., Gai, K.: Entire space multi-task model: An effective approach for estimating post-click conversion rate. In: SIGIR (2018)
21. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint (2018)
22. Ovaisi, Z., Ahsan, R., Zhang, Y., Vasilaky, K., Zheleva, E.: Correcting for selection bias in learning-to-rank systems. In: WWW (2020)
23. Pearl, J., Glymour, M.M., Jewell, N.P.: Causal inference in statistics: A primer (2016)

24. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: UAI. p. 452–461 (2009)
25. Saito, Y.: Asymmetric tri-training for debiasing missing-not-at-random explicit feedback. In: SIGIR (2020)
26. Saito, Y.: Unbiased pairwise learning from biased implicit feedback. In: SIGIR. pp. 5–12 (2020)
27. Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., Joachims, T.: Recommendations as treatments: Debiasing learning and evaluation. pp. 1670–1679. PMLR (2016)
28. Tao, W., Li, Y., Li, L., Chen, Z., Wen, H., Chen, P., Liang, T., Lu, Q.: Sminet: State-aware multi-aspect interests representation network for cold-start users recommendation. AAAI (2022)
29. Tran, T., Sweeney, R., Lee, K.: Adversarial mahalanobis distance-based attentive song recommender for automatic playlist continuation. In: SIGIR (2019)
30. Tran, T., You, D., Lee, K.: Quaternion-based self-attentive long short-term user preference encoding for recommendation. In: CIKM. pp. 1455–1464 (2020)
31. VanderWeele, T.J.: A three-way decomposition of a total effect into direct, indirect, and interactive effects. Epidemiology (2013)
32. Wang, W., Feng, F., He, X., Wang, X., Chua, T.S.: Deconfounded recommendation for alleviating bias amplification. In: KDD. pp. 1717–1725 (2021)
33. Wang, W., Feng, F., He, X., Zhang, H., Chua, T.S.: Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In: SIGIR (2021)
34. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: SIGIR. pp. 165–174 (2019)
35. Wang, X., Zhang, R., Sun, Y., Qi, J.: Doubly robust joint learning for recommendation on data missing not at random. In: ICML (2019)
36. Wei, C., Liang, J., Liu, D., Wang, F.: Contrastive graph structure learning via information bottleneck for recommendation. NeurIPS **35**, 20407–20420 (2022)
37. Wei, T., Feng, F., Chen, J., Wu, Z., Yi, J., He, X.: Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In: KDD (2021)
38. Xin, X., He, X., Zhang, Y., Zhang, Y., Jose, J.: Relational collaborative filtering: Modeling multiple item relations for recommendation. In: SIGIR. pp. 125–134 (2019)
39. Xu, S., Tan, J., Heinecke, S., Li, J., Zhang, Y.: Deconfounded causal collaborative filtering. arXiv preprint arXiv:2110.07122 (2021)
40. Yang, L., Cui, Y., Xuan, Y., Wang, C., Belongie, S., Estrin, D.: Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In: Recsys. pp. 279–287 (2018)
41. You, D., Vo, N., Lee, K., Liu, Q.: Attributed multi-relational attention network for fact-checking url recommendation. In: CIKM (2019)
42. Zhang, W., Bao, W., Liu, X.Y., Yang, K., Lin, Q., Wen, H., Ramezani, R.: Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning. In: WWW (2020)
43. Zhang, Y., Feng, F., He, X., Wei, T., Song, C., Ling, G., Zhang, Y.: Causal intervention for leveraging popularity bias in recommendation. In: SIGIR (2021)
44. Zhao, Z., Chen, J., Zhou, S., He, X., Cao, X., Zhang, F., Wu, W.: Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. TKDE (2022)
45. Zheng, Y., Gao, C., Li, X., He, X., Li, Y., Jin, D.: Disentangling user interest and conformity for recommendation with causal embedding. In: Proceedings of the Web Conference 2021. pp. 2980–2991 (2021)

46. Zheng, Y., Gao, C., Li, X., He, X., Li, Y., Jin, D.: Disentangling user interest and conformity for recommendation with causal embedding. In: WWW (2021)
47. Zhu, X., Zhang, Y., Feng, F., Yang, X., Wang, D., He, X.: Mitigating hidden confounding effects for causal recommendation. ArXiv (2022)
48. Zhu, Z., He, Y., Zhang, Y., Caverlee, J.: Unbiased implicit recommendation and propensity estimation via combinational joint learning. In: Recsys (2020)
49. Zhu, Z., He, Y., Zhao, X., Caverlee, J.: Popularity bias in dynamic recommendation. In: KDD (2021)